

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Urban Čopič

**Učinkovito oblikovanje izpisov v SAP ERP
informacijskem sistemu**

DIPLOMSKO DELO NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Ljubljana, 2016

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Urban Čopič

**Učinkovito oblikovanje izpisov v SAP ERP
informacijskem sistemu**

DIPLOMSKO DELO NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

MENTOR: viš. pred. dr. Igor Rožanc

Ljubljana, 2016

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco *GNU General Public License*, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses>.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

SAP ERP je znan po modularno sestavljeni programski rešitvi, s katero pokriva širok spekter poslovnih uporabnikov. Posledica je specifična arhitektura rešitev, ki se odraža v zahtevni izbiri potrebnih delov in dolgotrajnem prilagajanju sestavnih delov sistema specifičnim potrebam uporabnika. Tipičen primer je oblikovanje izpisov oziroma poročil.

V diplomski nalogi najprej predstavite osnove SAP-a s poudarkom na razvojnem okolju ABAP. V osrednjem delu opišite in na zgledu predstavite svojo rešitev za enostavno oblikovanje izpisov z orodjem Smartforms, ki omogoča bistveno učinkovitejše parametrizirano določanje izpisov. Nalogo zaključite s predstavitevijo prednosti in slabosti svoje rešitve.

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Urban Čopič,

z vpisno številko 63980212,

sem avtor/-ica diplomskega dela z naslovom:

Učinkovito oblikovanje izpisov v SAP ERP informacijskem sistemu

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)
viš. pred. dr. Igor Rožanc
in somentorstvom (naziv, ime in priimek)

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.)
ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 31.8.2016 Podpis avtorja/-ice: 

Na tem mestu se bi še posebej zahvalil svojemu mentorju dr. Igor Rožancu za koristne nasvete in podjetju Itelis d.o.o. za strokovno pomoč.

Kazalo

Povzetek

Abstract

1. Uvod	1
2. SAP osnove	3
2.1 Sistemsko okolje	6
2.2 SAP arhitektura	7
2.2.1 Plast podatkovne baze	7
2.2.2 Plast aplikacijskega strežnika	8
2.2.3 Plast predstavitvenega strežnika	8
2.2.4 Povezava med plastmi	8
2.3 ABAP razvojno okolje	10
2.3.1 ABAP skladiščenje objektov	11
2.3.2 ABAP razvojna orodja	13
2.3.3 ABAP programiranje	14
3. Razvoj prednastavljenih izpisov z orodjem Smartforms	17
3.1 SAP Smartforms okolje	18
3.1.1 Sestavni deli Smartform obrazca	19
3.2 Razvoj izpisa fakture	21
3.2.1 Razvoj okolja za vzdrževanje nastavitv izpisov	21
3.2.2 Sestavni deli fakturnega dokumenta	22
3.2.3 Lastne nastavitve izpisa	24
3.2.3.1 Splošne nastavitve	25
3.2.3.2 Nastavitve glave računa	28
3.2.3.3 Nastavitve postavke računa	30

3.2.3.4	Nastavitve noge računa.....	31
3.3	Primer izpisa fakture	33
3.4	Prednosti in slabosti rešitve.....	36
4.	Sklepne ugotovitve	39
5.	Viri in literatura	41

Seznam uporabljenih kratic

kratica	angleško	slovensko
SAP	Systems Applications Products	sistemi aplikacije proizvodi
ABAP	Advanced Business Application Development	napredni razvoj poslovnih aplikacij
ASAP	Accelerated SAP Methodology	metodologija pospešenega SAP
DEV	Development System	razvojni sistem
QA	Quality Assurance System	sistem zagotavljanja kakovosti
PROD	Production System	produkcijski sistem
DB	Database	podatkovna baza
CO	Controlling	nadzorni modul
FI	Financial Accounting	računovodstvo
HR	Human Resources	upravljanje s kadri
SD	Sales and Distribution	prodaja in distribucija
MM	Material Management	upravljanje z materialom
BC	Basis Module	modul osnove (administracija)
BI	Business Intelligence	poslovna inteligenca
BPC	Business Planning And Consolidation	planiranje procesov in konsolidacija
BODI	Business Object Data Integrator	integracija poslovnih objektov
EP	Enterprise Portal	portal podjetja
PO	Process Orchestration	nastavljanje procesov

Povzetek

Naslov: Učinkovito oblikovanje izpisov v SAP ERP informacijskem sistemu

Namen diplomskega dela je prikaz rešitve, ki poenostavlja implementacijo izpisov pri uvajanju SAP informacijskega sistema v podjetju. Izpisi so večkrat zanemarjena tema pri uvajanju, ker je vedno poudarek na popisu poslovnih procesov, pravilnem kreiranju matičnih podatkov in nastavitvah, kar je tudi pravilno, saj morajo biti podatki podjetja pravilno shranjeni in obdelani.

Na žalost, ob začetku uporabe informacijskega sistema, uporabniki vedno ocenjujejo kvaliteto implementacije po izgledu izpisov, ki so običajno končni rezultat oz. informacija odvijajočega procesa v SAP-u. Pogosto se dogaja, da za oblikovanje in kodiranje izpisov zmanjka časa in izpadejo iz proračunskega okvirja.

V diplomskem delu je predstavljen razvoj prednastavljenih SAP izpisov za prodajni modul, ki zajemajo najboljšo prakso iz poslovnega sveta. S prednastavljenimi izpisi je implementacija prodajnih izpisov parameterizirana in posegi ABAP programerja niso več potrebni. Izpise lahko nastavi vsebinski uvajalec.

Ključne besede: SAP, ERP, Smartforms, ABAP, implementacija

Abstract

Title: Efficient formatting of printouts in SAP ERP information system

The purpose of diploma thesis is to simplify the implementation of printouts in the project realisation phase of the SAP information system within a company.

Printouts are often a neglected subject in the initiation process, as in the beginning, the main focus is always on the inventory of commercial processes and on the accurate creation of master data and settings. This is of course understandable, because a company must ensure its data are properly stored and processed.

Unfortunately, at the beginning of the use of the information system, users tend to assess the quality of implementation by the appearance of prints, which is usually the final result or final piece of information of the running process in the SAP system. As far as the design and the coding of printouts is concerned, companies often run short of time, and therefore this part of the process is not included in the calculation of the budget frame.

This diploma thesis describes the development of preconfigured SAP printouts for a sales module, covering the best practices from the business world.

The implementation of sales printouts is parameterized by preconfigured printouts. Therefore, the intervention of an ABAP programmer is no longer needed. Outprints can be set by a SAP consultant.

Keywords: SAP, ERP, Smartforms, ABAP, implementation

1 Uvod

Ob implementaciji SAP rešitev, še posebej rešitev ki jih zajema ERP sistem se kot programerji vedno znova soočamo z implementacijo SAP izpisov. SAP izpisi so tisti dokumenti, ki se izpišejo na tiskalnik in so v fizični obliki namenjeni strankam. Primer SD prodajnih izpisov so recimo, izpis prodajnega naloga, izpis fakture ali izpis dobavnice.

SAP projekti so vodeni po ASAP metodologiji, ki zajema korake [1]:

- priprava projekta
- poslovni načrt
- realizacija
- končne priprave
- podpora ob prehodu v živo
- zagotavljanje delovanja in podpora

Okvirna oblika in vsebina izpisov se določi v koraku poslovnega načrta, v koraku implementacije pa sledi oblikovanje in kodiranje izpisa. Proces kodiranja se vedno znova ponavlja pri vsakem projektu in zahteva določen čas implementacije.

Predviden čas implementacije na posamezen izpis je približno 2-3 čd (človek dan). Če zahtevajo poslovni procesi implemetacijo 10 izpisov, to pomeni 30 čd razvoja, kar je za stranke včasih nesprejemljivo.

Ideja diplomske naloge je skrajšanje časa implementacije za vsaj 60%. To smo v diplomskem delu dosegli z prednastavljeno obliko in parameterizacijo podatkov na izpisih. S prednastavljeni izpisi ni potrebno ABAP kodiranja, ampak si lahko posamezni vsebinski uvajalec sam nastavi kateri podatki se bodo na izpisu pokazali.

Prednastavljeni izpisi se nahajajo v svojem razvojnem paketu. V procesu realizacije, ob namestitvi SAP sistemov, sistemski administrator naloži paket izpisov na sistem. Vsebinski

uvajalec med postopkom nastavljanja sistema poskrbi tudi za nastavitve izpisov in jih lahko med svojim delom tudi testira. ABAP programer je potreben samo v izjemnih primerih, če se stranka odloči za drugačen koncept izpisov. V tem primeru se tudi stroški projekta povečajo. Prednastavljena rešitev stranki omogoča odločitev ali bo vzela cenovno ugodno varianto rešitve ali pa implementacijo izpisov po svojih željah. V večini primerov se prednastavljeni izpisi implementirajo v majhnih in srednjih podjetjih.

Pozneje, ko podjetje razširi svoje poslovanje, se pojavi potreba po bolj dodelanih izpisih. To je tudi trenutek, ko je podjetje pripravljeno več investirati v svojo zunanjo podobo in se pojavi zahteva po bolj dodelanih izpisih.

Rešitev v diplomski nalogi zadošča zakonskim določbam, na primer pri izpisu fakture, ne predstavlja pa celostne podobe podjetja. Na izpisih so recimo barve vnaprej določene in se jih ne da spreminjati z nastavitvami.

Prvi del diplomskega dela smo posvetili opisu SAP okolja, da lahko bralec razume kako sistem tehnično deluje. Opisana je SAP arhitektura in delovanje ABAP programskega okolja. Ostali deli diplomskega dela so posvečeni opisi rešitve prednastavljenih izpisov. Prednastavljena rešitev ima poleg občutnih finančnih prihrankov tudi slabe strani. Te so opisane v zadnjem delu diplomskega dela.

2 SAP osnove

Začetki SAP-a segajo v leto 1972, ki ga je ustanovila skupina petih bivših IBM programerjev. Originalna SAP kratica izhaja iz nemškega jezika in pomeni Systeme Anwendungen Produkte, kar pomeni Sistemske Aplikacije in Proizvodi [2].

SAP je v 44 letih postal vodilna sila v svetu poslovnih aplikacij in z njimi povezanih storitvenih prihodkov. Na podlagi tržne kapitalizacije je SAP tretji največji neodvisni proizvajalec programske opreme na svetu. V 130 državah zaposluje 79.000 ljudi, storitve nudi 320.000 podjetjem v 190 državah sveta, sodeluje z 12.000 partnerskimi podjetji [2].

Začetna SAP arhitektura je bila imenovana *R/1*. Kratice R pomeni »Realtime«, 1 je oznaka za enonovijsko arhitekturo. Razvita je bila za potrebe financ in v ozadju ni imela baze. V poznih 70 letih je bila razvita dvonivojska arhitektura imenovana *R/2*. Delovala je na mainframe strežnikih. Zajemala je podatkovno bazo in aplikacijski strežnik. Prezentacijskega strežnika ni bilo. Na strani uporabnika je sistem deloval kot terminal [3].

Sodoben SAP je osnovana na trinivojski arhitekturi, imenovani *R/3* in zajema [4]:

- sloj podatkovne baze,
- sloj aplikacijskega strežnika in
- sloj predstavitvenega strežnika

Programi in podatki so shranjeni v podatkovni bazi. Aplikacijski strežnik poskrbi za izvajanje programov in obdelavo podatkov, oz. kot povezava med plastjo podatkovne baze in predstavitvenim strežnikom. Predstavitveni strežnik, imenovan tudi *SAPgui* je instaliran na uporabnikovi delovni postaji in služi interakciji z uporabnikom. Vsebina komunikacija med predstavitvenim in aplikacijskim slojem so podatki in kontrolni parametri. Kontrolni parametri definirajo izgled prikazanih podatkov in ekranov. Komunikacija je izredno učinkovita, saj so predloge ekranov shranjene v *SAPgui* instalaciji na uporabnikovi delovni postaji in se ne prenašajo po omrežni infrastrukturi. Z nastavitvami *SAPgui*-ja na posameznih delovnih postajah je mogoče nastavljanje različne izgleda ekranov oz. sloge. Uporabniška izkušnja

trinivojske arhitekture je kot da bi se vse odvijalo na delovni postaji uporabnika in ni občutka da se podatki obdelujejo na oddaljenih strežnikih.

Posamezne aplikacije v SAP-u se imenujejo transakcije in so namenjene tako uporabnikom kot razvijalcem. Transakcije so smiselno razporejene in služijo tako kreiranju matičnih podatkov, kreiranju dokumentov kot tudi razvoju. Razvijalci in uporabniki uporabljajo isto okolje, ki je ločeno z avtorizacijami, običajno na nivoju transakcij. V večjih poslovnih sistemih je potrebno definirati natančnejše avtorizacije, tudi na nivoju podatkov in organizacijske strukture.

SAP je razvil svoj programski jezik *ABAP* in svoje razvojno okolje *ABAP workbench*. *ABAP* kratica izhaja iz nemškega jezika, kar pomeni Allgemeiner Berichts Aufbereitungs Prozessor, v angleškem prevodu pomeni Advanced Business Application Development oz. v slovenščini Napredni Razvoj Poslovnih Aplikacij. Razvit je bil v 80 letih, kot del *R/2* dvonivojske arhitekture, z namenom razvoja poročil. *ABAP* sintaksa je enostavna, sodobno objektno orientirana, tako da se lahko razvijalec osredotoči na vsebino in se mu ni potrebno ukvarjati s samim programskim jezikom.

Priporočeno SAP sistemsko okolje je sestavljeno iz najmanj treh sistemov:

- *DEV* - razvojni sistem
- *QA* - sistem zagotavljanja kakovosti
- *PROD* - produkcijski sistem

Za majhna podjetja se lahko izjemoma uporabi dva sistema:

- *DEV/QA* razvojni sistem in sistem zagotavljanja kakovosti
- *PROD* produkcijski sistem

Razvojni sistem služi razvijalcem in vsebinskim svetovalcem. Uporabniki do tega sistema nimajo dostopa. **Sistem zagotavljanja kakovosti** služi testiranju uporabnikov. Ponavadi je stara kopija produkcije. **Produkcijski sistem** služi rednemu delu in je pod budnim očesom systemskega administratorja. Na sistemu zagotavljanja kakovosti in produkciji programiranje in nastavitve niso dovoljene. Vsi vpleteni morajo slediti protokolu prenašanja nastavitvev in programov na produkcijo. Nastavitve in programi se po sistemskem okolju prenašajo v vrstnem redu *DEV-QA-PROD*. Spremembe se prenašajo s pomočjo *Transport Request*-ov. V ta namen se uporablja orodje *TMS Transport Managment System*. Tehnično gledano so *Transport Requesti* datoteke shranjene v mapi v skupni rabi, ki je vidna vsem aplikacijskim strežnikom.

SAP rešitve so integrirane med seboj znotraj modulov. Moduli so razdeljeni v tri glavne skupine:

- funkcijski moduli
- tehnični moduli
- moduli specifični za industrijo

Najpogosteje uporabljeni in razširjeni funkcijski moduli v slovenskih podjetjih:

- *CO Controlling* – nadzorni modul
- *FI Financial Accounting* – računovodstvo
- *HR Human Resources* – upravljanje s kadri
- *SD Sales and Distribution* – prodaja
- *PP Production Planning* – planiranje proizvodnje
- *MM Material Management* – upravljanje z materialom

Glavni tehnični moduli:

- *ABAP modul* - programiranje
- *BC Basis module* – SAP administracija
- *BI Business Intelligence* – poslovna inteligenca
- *BPC Business Planning and Consolidation* – planiranje procesov
- *BODI module* – integracija poslovnih objektov
- *EP Enterprise Portal* – portal podjetja
- *Solution Manager* – upravljanje z rešitvami
- *PO Process Orchestration* – nastavljanje procesov

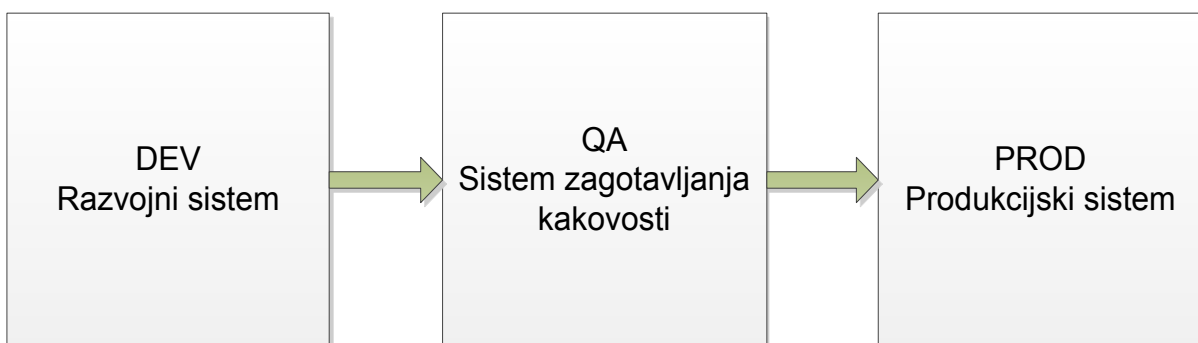
Primer modulov specifičnih za industrijo:

- *IS Aerospace & Defense* – obrambno ministrstvo in letalstvo
- *IS Telecommunications* - telekomunikacije
- *IS Defense & Security* – rešitev za obrambno ministrstvo in varnost
- *IS Higher Education & Research* – rešitev za univerze
- *IS Public Sector* – rešitev za javni sektor
- *IS Insurance* – zavarovalniška rešitev

Funkcijski in tehnični moduli so privzeti ob SAP namestitvi. Module specifične za industrijo je potrebno naknadno namestiti na sistem in so vidni kot *SAP ADD-IN*. V namen instalacije se uporablja orodje oz. transakcija *SAINT*.

2.1 Sistemsko okolje

Tipična SAP arhitektura je trinivojska. Sestavljena je iz razvojnega sistema, sistema zagotavljanja kakovosti in produkcije (slika 1).



Slika 1: Tri-nivojska arhitektura

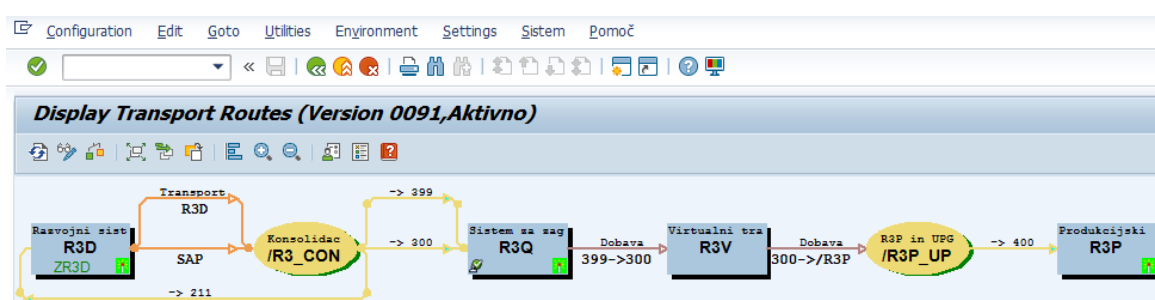
Ideja trinivojske postavitve sistemov je zagotavljanje konsistentnosti nastavitvev in *ABAP* programov.

DEV razvojni sistem služi razvijalcem in vsebinskim svetovalcem za nastavljanje sistema glede na potrebe podjetja. V začetni fazi implementacije projekta, so vse aktivnosti svetovalcev usmerjene v nastavitve razvojnega sistema. Ko so nastavitve in razvoj končane so spremembe lansirane s *transport requesti* na *QA* sistem zagotavljanja kakovosti.

QA sistem služi testiranju novih funkcionalnosti, ki so bile prenesene iz razvoja. Če se ob testiranju pokažejo napake, je potrebno nove nastavitve znova prenesti iz razvojnega sistema. Neposredno nastavljanje *QA* sistema ni dovoljeno.

PROD produkcijski sistem je namenjen rednemu delu uporabnikov. Pred »prehodom v živo« novega projekta, uporaba novih funkcionalnosti ni dovoljena, čeprav je sistem mogoče že nastavljen.

Prenos nastavitvev med razvojnim sistemom in produkcijskim sistemom je omogočen z SAP orodjem *TMS* (angl. Transport Managment System). V *TMS* nastavitvah so definirane transportne poti, po katerih se transportni zahtevki gibljejo (slika 2).



Slika 2: TMS transportne poti

2.2 SAP arhitektura

SAP deluje na trinivojski arhitekturi imenovani *R/3*, ki jo sestavljajo:

- plast podatkovne baze,
- plast aplikacijskega strežnika in
- plast predstavitvenega strežnika

2.2.1 Plast podatkovne baze

Plast podatkovne baze je sestavljena iz centralnega DB sistema, v katerem so hranjeni vsi *R/3* podatki. Na primer, baza vsebuje kontrolne podatke in nastavitve, ki definirajo kako SAP deluje. Vsebuje tudi programsko kodo aplikacij. Poleg programske kode so aplikacije sestavljene tudi iz definicije ekranov, menujev in raznih drugih komponent. Shranjene so v posebnem predelu baze, ki se imenuje *R/3* skladišče (angl. *R/3 Repository*).

DB sistem je sestavljen iz dveh komponent, DBMS (angl. *Database Managment System*) in baze same. V preteklosti je SAP uporabljal različne baze (MSSQL, Oracle, MaxDB). Ker pa je bil odvisen od različnih proizvajalcev je razvil svojo bazo SAP HANA. Posebnost SAP HANA je ta, da se ob zagonu baze najpogosteje uporabljeni podatki (celotne tabele) naložijo v fizični pomnilnik (RAM). S tem se izbor podatkov pohitri do te mere, da indeksi na tabelah niso več potrebni. Najmanjša zahteva po RAM pomnilniku je zato 128GB. Glede na velikost baze je tudi potreba po RAM-u ustrezno velika. Male instalacije imajo povprečno v produkcijskem okolju 512GB RAM, velika podjetja pa od 1TB RAM-a naprej.

2.2.2 Plast aplikacijskega strežnika

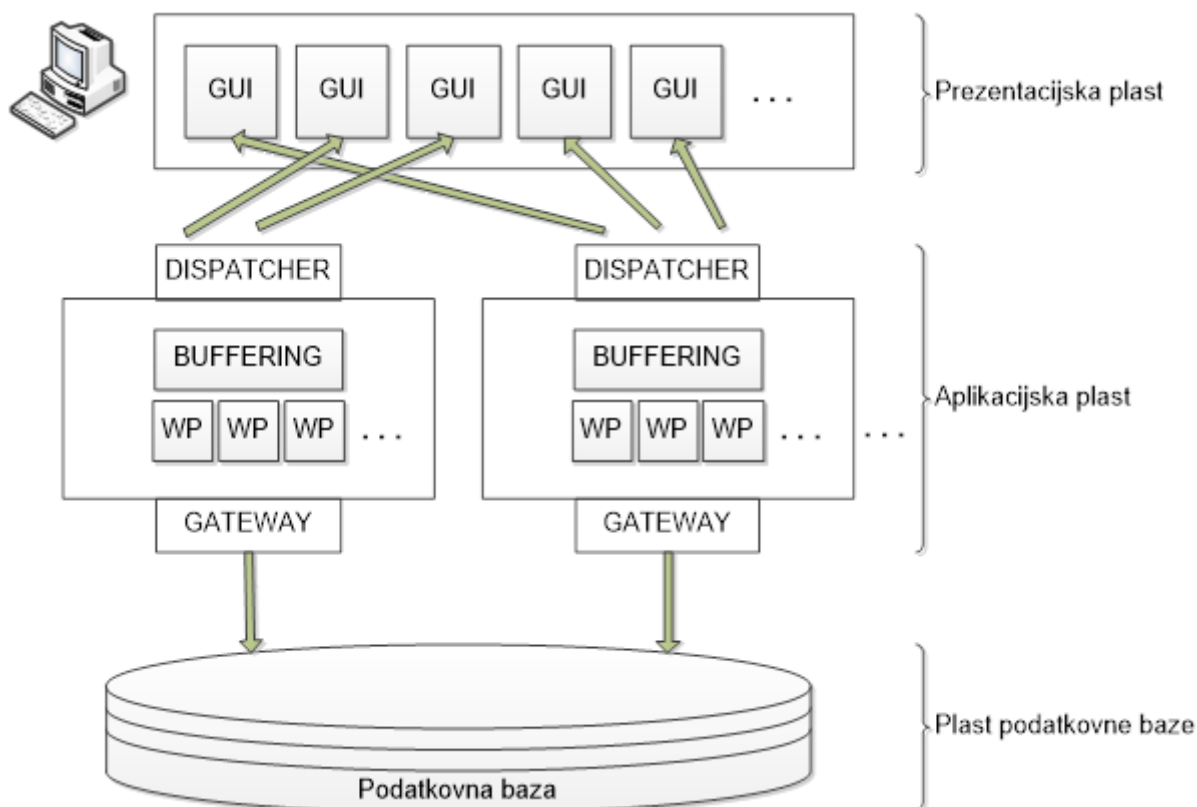
Sestavljena je iz enega ali več aplikacijskega strežnika. Vsak aplikacijski strežnik vsebuje nabor storitev, ki so potrebne da SAP normalno deluje. Storitve so porazdeljene po večjih aplikacijskih strežnikih. Za upravljanje storitev skrbi strežnik sporočil (angl. *Message server*), ki je tudi del plasti aplikacijskega strežnika. Strežnik sporočil skrbi za komunikacijo med aplikacijskimi strežniki. Znotraj sistema podaja zahtevke od enega strežnika do drugega. Vsebuje tudi podatke o aplikacijskih skupinah, ki služijo porazdelitvi obremenitve (angl. *Load balancing*) aplikacijskih strežnikov. Porazdelitev obremenitev se izvede ob prijavi uporabnika v sistem. Glede na odzivne čase aplikacijskih strežnikov je uporabnik preusmerjen na tisto instanco, ki ima najkrajši odzivni čas [4].

2.2.3 Plast predstavitvenega strežnika

Služi interakciji med uporabnikov in SAP sistemom. Uporablja se *SAPgui* (angl. *SAP Graphical User Interface*), ki je instaliran na uporabnikovi delovni postaji. Predstavitevna plast komunicira z aplikacijskim strežnikom in mu posreduje in vrača podatke. Uporabniku podatke prikaže na zaslonu v obliki ekranov in vnosnih polj. *SAPgui* je primarno napisan za Windows okolje. Obstaja tudi za druge operacijske sisteme (Java verzija), vendar z omejenim delovanjem [4].

2.2.4 Povezava med plastmi

Tehnična distribucija programov je neodvisna od lokacije strojne opreme, na kateri se nahaja. Vse plasti trinivojske arhitekture so lahko nameščene ena na drugo na enem strežniku, lahko pa je vsaka plast na svojem strežniku (slika 3) [4].



Slika 3: Komunikacija med posameznimi plastmi

Horizontalno se predstavitevni in aplikacijski strežnik lahko razdeli na več računalnikov. Horizontalna distribucija plasti podatkovne baze pa je odvisna od tipa nameščene baze.

Dispečer (anlg. Dispatcher): deluje kot vmesnik med predstavitveno plastjo in aplikacijsko plastjo. Zahteve podane od uporabnika preidejo preko dispečerja na aplikacijsko plast. Ko je zahteva obdelana se vrne nazaj do uporabnika preko dispečerja.

Prehod (angl. Gateway): vmesnik povezuje aplikacijsko plast s plastjo podatkovne baze. Preko prehoda se prenašajo zahteve za brisanje/vstavljanje/branje/spremembo nad podatkovno bazo.

Predpomnilnik (angl. Buffer): v predpomnilniku so začasno shranjeni najpogostejše uporabljeni podatki iz podatkovne baze, kot so podatki o avtorizacijah, nastavitve procesov, podatki o uporabnikih. Predpomnilnik se prilagaja glede na velikost pomnilnika aplikacijskega strežnika in se periodično sinhronizira. S predpomnilnikom dosežemo boljše odzivne čase, saj podatek prebere hitreje kot iz podatkovne baze.

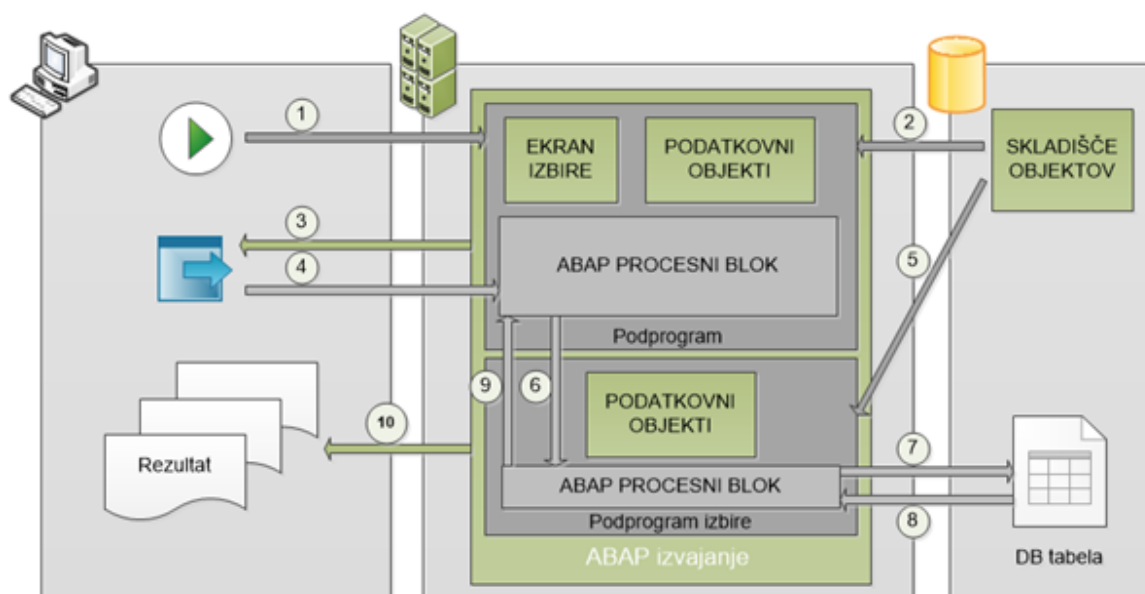
Delovni procesi (angl. Work Process): delovni proces je področje transakcijske obdelave podatkov. Poznamo pet vrst delovnih procesov.

- **Dialogni (angl. Dialog):** obdeluje zahteve aktivnih uporabnikov oz. posamezne klik-e uporabnika v *SAPgui*.
- **Update:** izvaja zahteve po spremembi nad bazo podatkov
- **Obdelava v ozadju (angl. Background):** služi izvajanju opravil v ozadju, na aplikacijskem strežniku, tudi ko uporabnik ni prijavljen v sistem preko *SAPgui*.
- **Proces zaklepanja (angl. Enqueue):** proces krmili tabelo zaklepanj (angl. *Lock Table*), ki se nahaja v pomnilniku v skupni rabi, ki je viden vsem aplikacijskim strežnikom. Proces zaklepanja je lahko samo eden.
- **Spool process:** tiskalniku posreduje zahteve tiskanja in jih shrani v čakalno vrsto. Zahteve tiskanja so oštevilčene v vrtnem redu kreiranja.

2.3 ABAP razvojno okolje

ABAP programi se izvajajo na aplikacijskem strežniku. SAP sistem predstavlja povprečnemu uporabniku »črno škatlo«, saj mu ni potrebno razumeti natančnega toka programa skozi plasti. Zanima ga samo poslovni proces in kako naj prikaže in vnese podatke. Tehnični vidik programa za uporabnika ni pomemben.

Po drugi strani morajo razvijalci, če hočejo razvijati svoje lastne programe, razumeti interakcijo med nivoji strežnika in poslovnimi procesi (slika 4) [5].



Slika 4: Primer izvajanja ABAP poročila

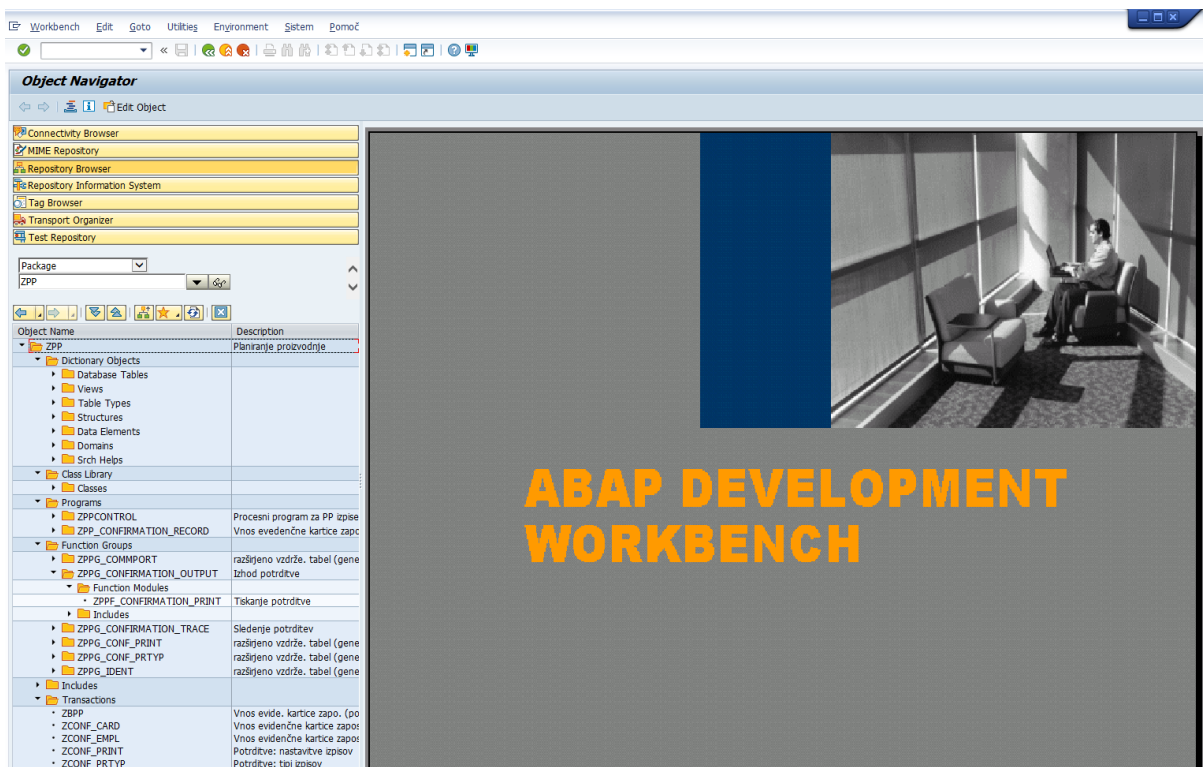
Uporabnik zažene transakcijo iz SAP menuja (1). Kontekst programa vsebuje področja spomina, ki služijo hranjenju spremenljivk in kompleksnih podatkovnih tipov. Vse te informacije o programu pridobi ABAP okolje iz skladišča objektov (2), ki je shranjen v posebnem delu baze podatkov. Ker program vsebuje ekran izbire, ABAP okolje posreduje ekran izbire predstavitevniemu strežniku (3). Predstavitevni strežnik nadzoruje tok programa toliko časa, dokler se uporabnik ne odloči, da je zaključil z vnosom podatkov v vnosna polja ekrana izbire [5].

Ko uporabnik zaključi z vnosom podatkov na ekran izbire pritisne gumb izvajanje. Vneseni podatki se prenesejo v ustrezne podatkovne objekte v programu in ABAP izvajanje prevzame nadzor (4). Procesna rutina ABAP procesnega bloka zajame potrebne podatkovne objekte iz baze (5) in jih naloži na aplikacijski strežnik. Izvajanje med ekranom izbire in DB izvajanjem je sinhrono, kar pomeni da procesna rutina selekcijskega ekrana počaka na DB izbiro procesne rutine DB izvajanja (6). Baza vrne zahtevane podatke ABAP procesnemu bloku (7, 8), ki jih razporedi po ustreznih podatkovnih objektih [5].

Procesni blok podprograma izbire vrne podatke podprogramu za obdelavo podatkov (9). ABAP okolje vrne rezultat prezentacijskemu strežniku [5].

2.3.1 ABAP skladiščenje objektov

ABAP skladišče objektov je sestavljeno iz razvojnih objektov, kot so programi, funkcijski moduli definicija DB tabel. V skladišču objektov so večinoma SAP standardni objekti, kot tudi objekti lastnega razvoja. Objekti so shranjeni v bazi. To pomeni, da do njih lahko dostopa vsaki klient in so vsem enaki. Podatkovna baza vsebuje tudi aplikacijske podatke in podatke nastavitvev. Objekti so organizirani glede na komponente aplikacij. Recimo modul PP (angl. Production Planning) je sestavljen iz več paketov, ki vsebujejo logiko za krmiljenje produkcijskih procesov (slika 5, primer hierarhije objektov v paketu). Nekateri objekti so sestavljeni iz podobjektov in se lahko sklicujejo na druge objekte.



Slika 5: Hierarhija objektov znotraj paketa

Najpogosteje uporabljeni objekti v skladišču [6]:

Paket (angl. Package): služi združitvi skupine objektov, ki si delijo očitne lastnosti, primer: programi (in vse njegove komponente) za izpis odprtih postavk kupcev in dobaviteljev v FI modulu se nahajajo v svojem paketu.

Podatkovni element (angl. Data Element): polje definirano v tabeli ali strukturi je dodeljeno podatkovnemu elementu. Uporablja se tudi recimo kot referenco na podatkovni element pri definiciji spremenljivk v ABAP programu.

Domena (angl. Domain): definira dolžino in tip podatkovnega elementa.

Struktura (angl. Structure): struktura je sestavljena iz komponent, ki so lahko polja, vključene strukture ali tip tabele. Strukture se lahko tudi uporabljajo kot referenca na polja pri razvoju ekranov in kot interne strukture znotraj ABAP programa.

Tabela (angl. Table): tabela se kreira neodvisno od tipa podatkovne baze. Tabele kreirane v ABAP slovarju so avtomatično prevedene v format, ki ga podatkovna baza podpira. Sestavljene so iz polj, ki imajo referenco na podatkovni element.

Tip tabele (angl. Table Type): tip tabele je definicija interne tabele znotraj ABAP programa in obstaja dokler se program izvaja. Sestavljena je iz strukture.

Program: izvedljiva koda ABAP programa, ki je lahko tudi vključljiv program. Vključljiv program je del glavnega programa in se ga ne da neposredno izvajati.

Vključljiv program (angl. Include Program): uporablja se za modulariziranje izvirne kode v glavnem programu. Omogoča nam, da isto ABAP kodo uporabljamo v različnih glavnih programih.

Funkcijski modul (angl. Function Module): funkcijski moduli so podprogrami, ki jih lahko kličemo v različnih programih. Sestavljeni so iz vhodnih in izhodnih parametrov. Združeni so znotraj bazena modulov (angl. Module Pool). Funkcijski moduli v istem bazenu modulov si lahko delijo skupne globalne spremenljivke. Razlika med vključljivimi programi in funkcijskimi moduli je ta, da funkcijske module lahko neodvisno izvajamo in testiramo.

Razred (angl. Class): razred predstavlja abstraktni opis objekta oz. predstavlja navodila za graditev objekta. Atributi objekta so predstavljeni s komponentami razredam, ki opisujejo obnašanje objekta.

2.3.2 ABAP razvojna orodja

ABAP razvojna orodja [7] se uporabljajo za vzdrževanje objektov v skladišču (angl. *Repository Objects*). Ta orodja pokrivajo celotni softverski razvojni cikel.

ABAP vzdrževanje (angl. ABAP Editor): služi kreiranju in vzdrževanju ABAP kode.

ABAP slovar (angl. Dictionary): služi vzdrževanju definicij tabel, podatkovnih tipov in ostalih entitet.

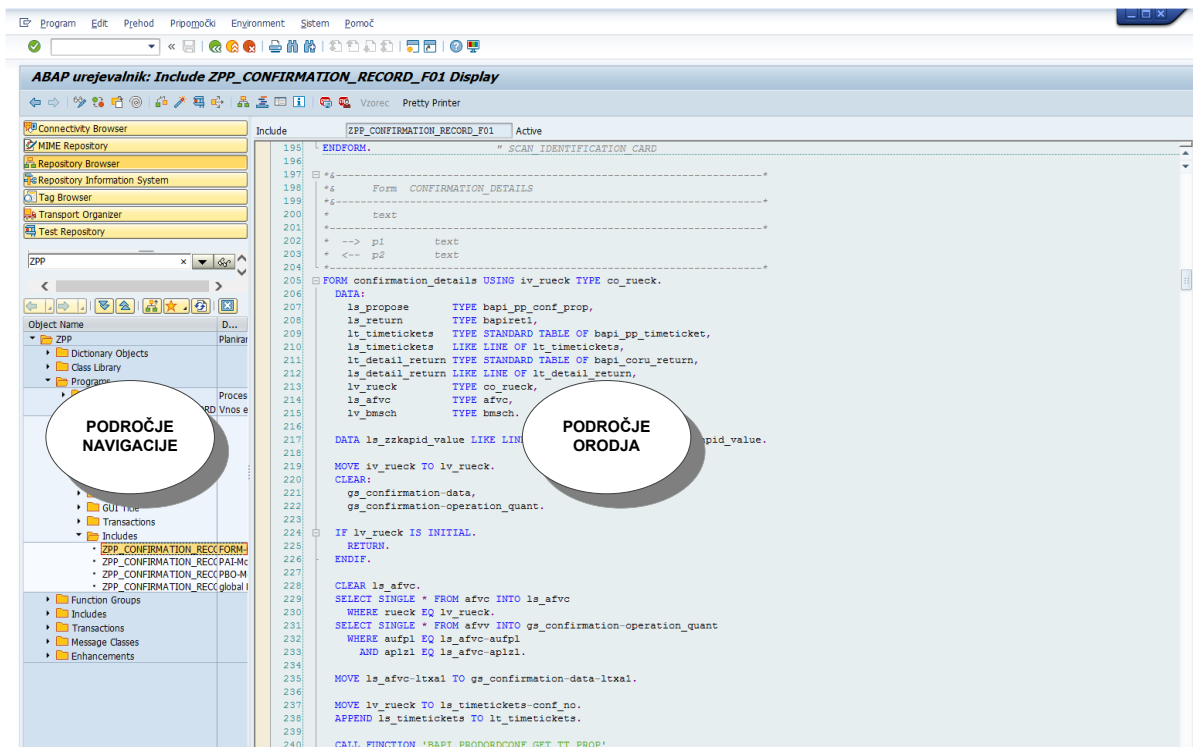
Razvoj ekranov (angl. Screen Painter): vzdrževanje ekranov skupaj s funkcijami za dialog z uporabniki.

Razvoj menuja (angl. Menu Painter): služi razvoju in vzdrževanju komponent uporabniških vmesnikov kot so vrstica menuja, orodne vrstice, funkcijske tipe.

Razvoj funkcijskih modulov (angl. Function Builder): namenjen vzdrževanju funkcijskih modulov in bazenov modulov.

Razvoj razredov (angl. Class Builder): kreiranje definicij razredov metod in atributov.

Vsa našeta razvojna orodja so združena v centralnem razvojnem orodju (angl. *ABAP Development Workbench*). *ABAP Development Workbench* (slika 6) je razdeljen na področje navigacije in področje orodja (angl. Tool Area).



Slika 6: ABAP razvojno okolje

2.3.3 ABAP programiranje

Vsi objekti lastnega razvoja morajo biti v svojem imenskem področju (angl. Namespace). S svojim imenskim področjem se zagotovi ločitev med svojim lastnim razvojem in SAP standardnim razvojem. S tem preprečimo, da se ob SAP-ovi distribuciji novih funkcionalnosti, ne prepišejo lastni programi ali tabele, v primeru, da se po naključju SAP odloči za isto poimenovanje[8].

Svoje lastne rešitve lahko prenesemo tudi večim strankam. V primeru da imamo registrirano svoje lastno imensko področje, stranke ne morejo popravljati ABAP kode. ABAP kodo se lahko na ne-originalnem sistemu popravlja le, če registriramo spremembo na SAP podpornem portalu.

Imensko področje predstavlja prvih nekaj znakov imena. Stranka oz. lastnik sistema si lahko registrira svoje imensko področje, ki se začne in konča z znakom /. Imensko področje se registrira na SAP support portalu in ga ne sme uporabljati nobena od strank.

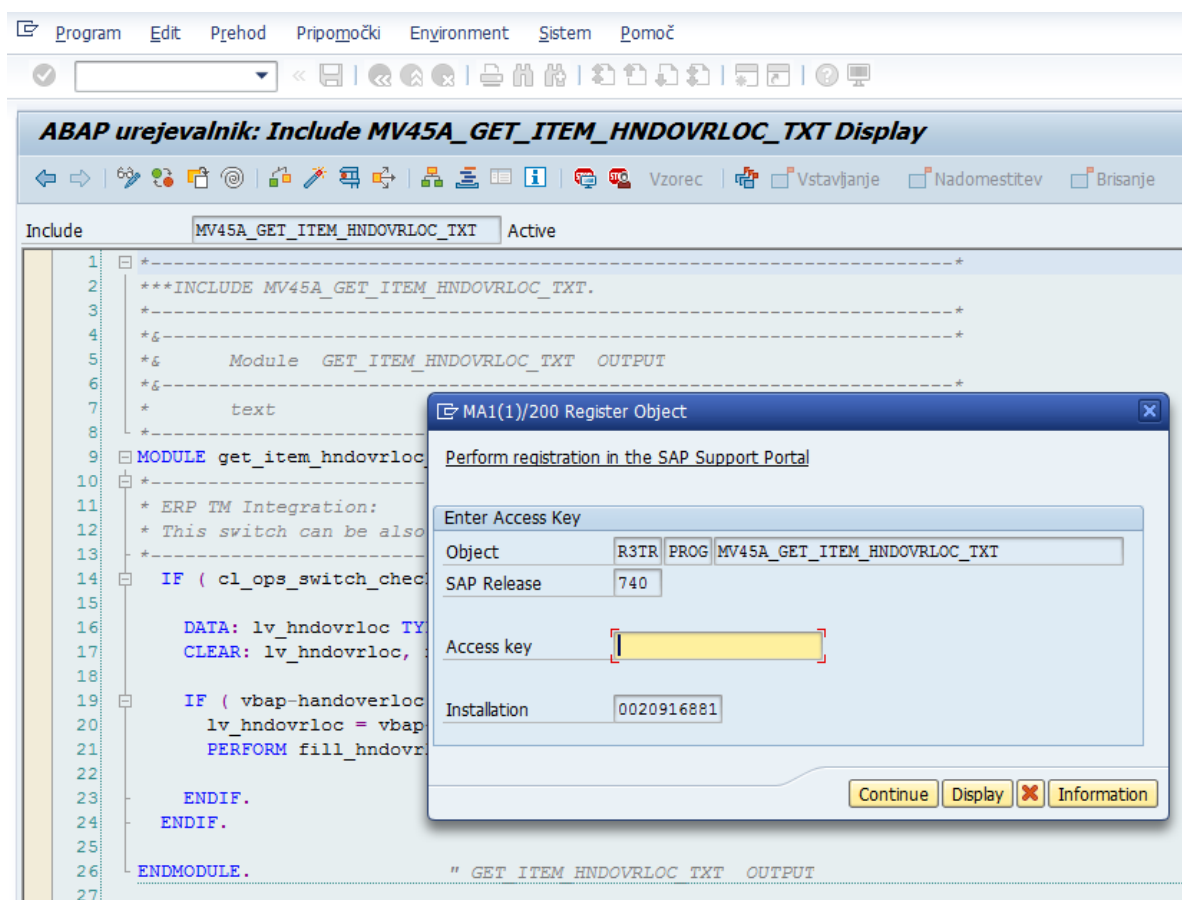
Primer imena programa za pregled odprtih postavk dobaviteljev /FRI/VEND_OPEN_ITM. V tem primeru bi bil /FRI/ registrirano imensko področje.

Če imensko področje ni registrirano, se uporablja začetna črka imena Z ali Y, recimo ZVEND_OPEN_ITM.

ABAP gradnike ločimo po izvoru na:

- SAP standardni razvoj in
- lastni razvoj v svojem imenskem področju

SAP standardni razvoj ni dovoljeno spreminjati. Spreminjamo ga lahko le v primeru, ko so na *SAP support portal-u* lansirana navodila za odpravo napake. Opis in odpravo napake imenujemo *SAP note*. Za spremembo SAP standarda potrebujemo dostopni ključ (angl. Access Key), ki ga pridobimo na SAP portalu. Vpišemo ga ob prvem kliku na ikono spremembe (slika 7).



Slika 7: Registracija spremembe SAP standardnega programa

3 Razvoj prednastavljenih prodajnih izpisov z orodjem Smartforms

SAP izpisi temeljijo na standardnih nastavitvah izhodov (angl. *output*) in so sestavljeni iz procesnega programa in *smartform*. Na posameznih izhodih je implementirana logika, ki definira izgled in podatke izpisa. Podatki izpisa so pridobljeni s procesnim programom, izgled izpisa pa definira obrazec *smartform*.

Nastavitve, ki so potrebne pri določanju izhoda:

- **vrsta sporočila** – določa posamezne izpise znotraj aplikacije, recimo znotraj prodajnega naloga določa ponudbo, prodajni nalog
- **aplikacija** – določa v kateremu prodajnem področju je izpis namenjen
- **prenosni medij** – tiskalnik, EDI, e-pošta
- **program** – implementirana logika za iskanje podatkov uporabljenih na smartform-i
- **form rutina** – začetna rutina, ki se pokliče v procesnem programu. Procesni program ima lahko različne začetne routine. Posledično lahko isti procesni program uporabimo za različne izpise le da so začetne routine drugačne.
- **smartform obrazec** – obrazec, ki služi oblikovanju izgleda

Posamezni izhodi prodajnih izpisov so razdeljeni na aplikacije, ki definirajo področje uporabe izhoda [9]:

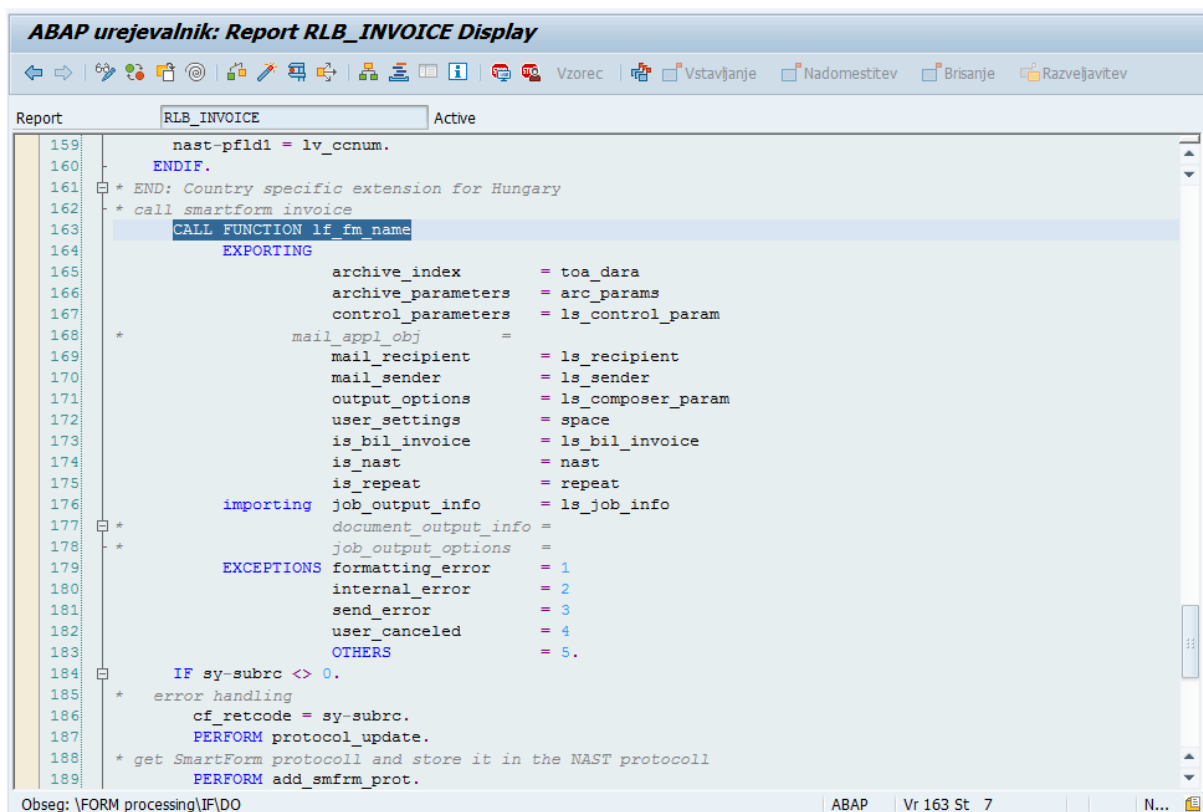
- *V1* – prodajni nalog,
- *V2* – odprema in
- *V3* – fakturiranje

V diplomskem delu so bile uporabljene SAP standardne nastavitve, recimo uporabljen je bil standardni procesni program za iskanje podatkov. Lastni razvoj in parameterizacija je bila uporabljena le na *smartform* obrazcu.

3.1 SAP Smartforms okolje

Orodje za razvoj izpisov je sestavljeno iz treh delov (slika 8):

Obrazec: *Smartform* obrazec vsebuje vsa potrebna navodila za generiranje izpisa. Sestavljen je iz sloga, strani, oken in logike iskanja podatkov. Razvijalec na obrazcu definira obliko in logiko prikaza, ob shranjevanju in generiranju obrazca se samodejno kreira *ABAP* funkcijski modul obrazca. Funkcijski modul obrazca se kliče v *ABAP* procesnem programu z ukazom `CALL FUNCTION` (slika 8) [10].



Slika 8: klicanje funkcijskega modula izpisa

Slog: slog definira obliko teksta v *smartform* obrazcu in modulu teksta. Isti slog lahko uporabljamo v večih obrazcih in modulih teksta.

Modul teksta: tekstovni moduli so namenjeni centralnemu shranjevanju teksta, ki ga lahko uporabljamo na različnih izpisih. Posamezni modul se kliče na *smartform* z imenom in jezikom

prikaza. Oblika teksta je definirana s slogom, ki ga vpišemo v glavo teksta. Na posameznih vrsticah in odstavkih določimo format odstavka in oblike znakov, ki jih črpamo iz sloga v glavi.

3.1.1 Sestavni deli Smartform obrazca

Smartform obrazec je sestavljen iz globalnih nastavitev, strani in oken.

Globalne nastavitve:

Atributi obrazca: določajo format strani in slog

Vmesnik obrazca: definirani so uvozni parametri generiranega funkcijskega modula. Procesni program preko vmesnika obrazca poda podatke na obrazec. Podatke lahko na obrazcu še dodatno obdelujemo.

Globalne definicije: v globalnih definicijah določimo globalne spremenljivke, strukture in tabele. Spremenljivke so vidne celotnemu smartform izpisu.

Strani in okna [11]:

Stran: določa obliko strani in spada na najvišji nivo hierarhije v obrazcu. Vsi objekti, ki se v formi uporabljajo, so podrejeni strani. Vsaka stran ima povezavo na naslednjo stran. Stran lahko kaže tudi sama nase.

Okno: okno je definirano znotraj strani in predstavlja zaokrožen predel obrazca. Znotraj okna definiramo nadaljna navodila izpisa. Poznamo več vrst oken, najpomembnejša so: zaključno okno, glavno okno, stransko okno. Razlikujejo je po času izvajanja. Glavno okno ima to lastnost, da preskoči na naslednjo stran, ko je zapolnjeno z vsebino.

Grafika: služi prikazu slik. Definirajo jo atributi: ime, objekt, ID, globina barv (črnobelo, barvno), resolucija.

Naslov: služi izpisu formatiranega naslova. Elementu naslova dodelimo številko naslova (angl. *Address Number*). Sistem zna sam, glede na državo in jezik izpisa oblikovati naslov v pravilnem formatu in jeziku.

Tekst: namenjen je dodajanju teksta na obrazec. Tekst se lahko nahaja na sami formi, lahko pa ga vključimo iz predhodno shranjenega teksta, recimo tekst modula.

Tabela: prikaz podatkov v obliki tabele. Tabela je sestavljena iz treh področij: glava tabele, postavke tabele, noga tabele. Tabela se dinamično prilagaja tekstu v vertikalni smeri. Ko tekst

doseže levi ali desni rob celice preskoči v novo vrstico. Primer uporabe je izpis računa: glava z nazivi stolpcev, postavke s ceno, končne vrednosti v nogi tabele.

Šablona: podobno tabeli, definirane so vrstice in stolpci. Od tabele se razlikuje po temu, da se šablona ne prilagaja tekstu. Ko tekst doseže rob celice se ne prikazuje več. Primer uporabe je recimo pri tiskanju položnic, kjer je potrebno tekst zapolniti v prednatisnjena okna.

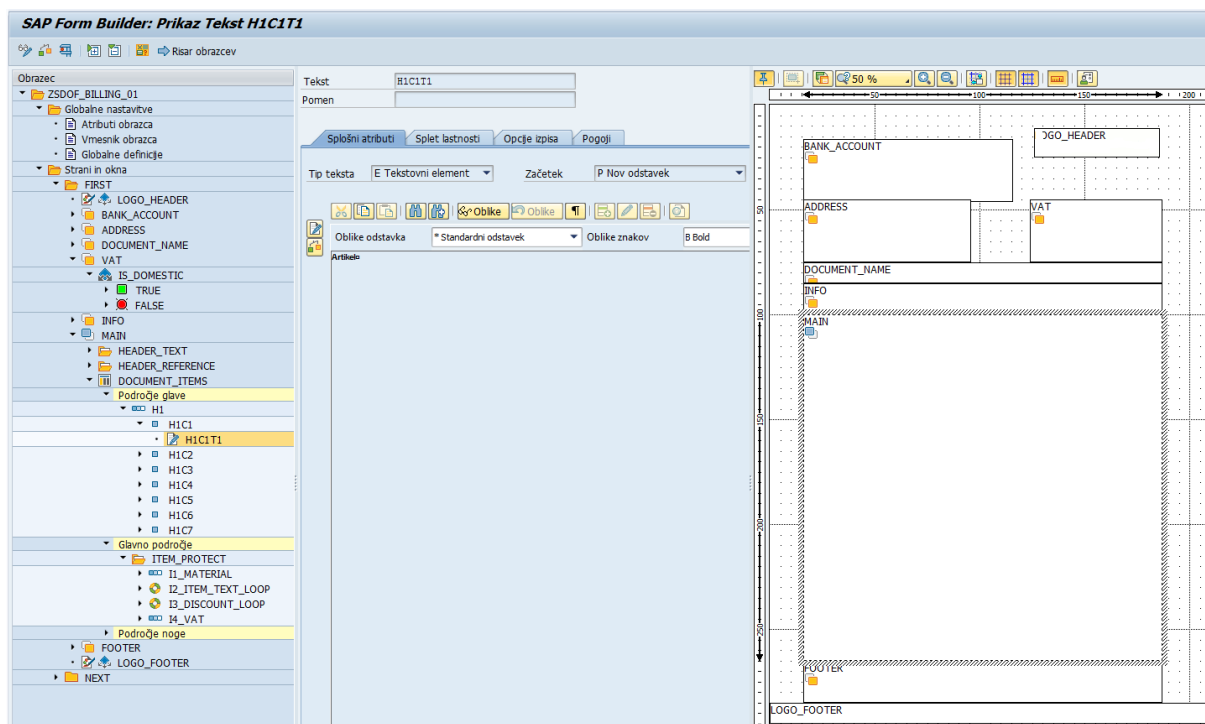
Mapa: ne vpliva na izgled obrazca. Služi združevanju posameznih elementov da je forma preglednejša.

Alternativa: vrne rezultat `true/false`. Uporabljamo jo, ko želimo izvajanje obrazca preusmeriti na dva dela.

Vrstica programa: *ABAP* koda na obrazcu. Globalne spremenljivke v kodi niso vidne. Podati jih moramo z vhodnimi in izhodnimi parametri. V generiranemu funkcijskemu modulu obrazca je vrstica programa vidna kot form rutina.

Ukaz: običajno se uporablja za nadzorovan prehod na novo stran, običajno v glavnem oknu.

Orodje *smartform* je sestavljeno iz treh področij (slika 9). Na levi strani je drevesna struktura elementov obrazca, na sredinskem področju so atributi in vsebina posameznih elementov, na desni strani je predogled pozicije oken.



Slika 9: Smartform orodje

Smartform obrazec je lahko vzdrževan v več izbranih jezikih. Jezik, v katerega smo prijavljeni v sistem ob kreiranju obrazca, postane privzeti jezik. V vse ostale jezike je potrebno obrazec prevesti z urejevalnikom prevajanja. V katerem jeziku se bo obrazec prikazal je odvisno od nastavitve komunikacijskega jezika kupca v matičnih podatkih.

3.2 Razvoj izpisa fakture

Poznamo več vrst fakturnih dokumentov:

- račun,
- storno računa,
- dobropis,
- storno dobropisa,
- bremepis,
- storno bremepisa,
- predračun in
- avansni račun

Vsak izmed dokumentov ima svoje lastne značilnosti, ki jih bi lahko opisali s posameznimi *smartform* obrazci. Ker pa imajo vsi dokumenti skupno obliko, je bila razvit en *smartform* obrazec za vse dokumente. S tem smo poenostavili vzdrževanje obrazcev. Običajno bi znotraj obrazca logiko za izpis posameznih dokumentov fiksno zapisali, v tej rešitvi so vse značilnosti parameterizirane. Nastavitve se nahajajo v lastnih tabelah in se berejo na obrazcu.

3.2.1 Razvoj okolja za vzdrževanje nastavitve izpisov

Vzdrževanje nastavitve izpisov je bilo narejeno z SAP vzdrževalcem tabel (angl. *Table maintenance*)[12]. To je orodje, ki samodejno kreira ekrane vzdrževanja na podlagi kreirane lastne tabele. Orodje samodejno kreira *ABAP* kodo, ki poskrbi za:

- prikaz vnosov,
- kreiranje vnosov,

- spremembo vnosov,
- logiko zaklepanja,
- funkcije pomikanja po seznamu (stran gor, stran dol, pomik levo, pomik desno),
- shranjevanje vnosov in
- brisanje vnosov

Samodejna *ABAP* koda je kreirana v funkcijskem modulu, znotraj funkcijske skupine, ki jo je potrebno predhodno definirati v svojem lastnem imenskem področju. *Table maintenance* nastavitve so definirane na način, da je uvajalec prisiljen slediti zaporedju prenosa nastavitvev: začetne nastavitve na *DEV* sistemu, prenos nastavitvev na *QA* sistem in testiranje, prenos nastavitvev na *PROD* produkcijski sistem.

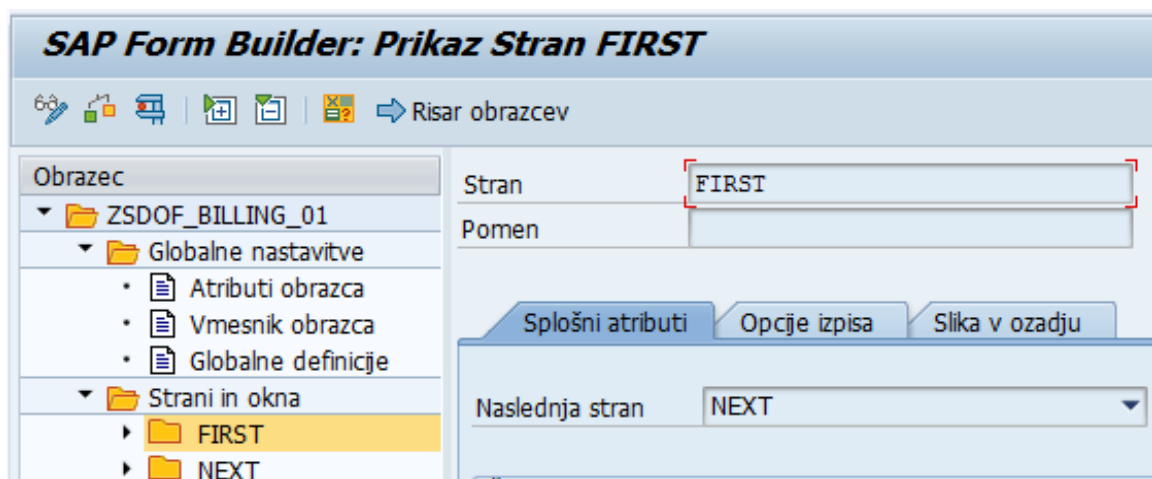
3.2.2 Sestavni deli fakturnega dokumenta

Fakturni dokument smo sestavili iz naslednjih področij:

- glava
 - logotip
 - bančni račun
 - naslov kupca
 - naziv dokumenta
 - davčna številka
 - informacija o kupcu
- osrednji del
 - glava postavk
 - postavke
 - končni izračuni

- zaključne klavzule
- noga

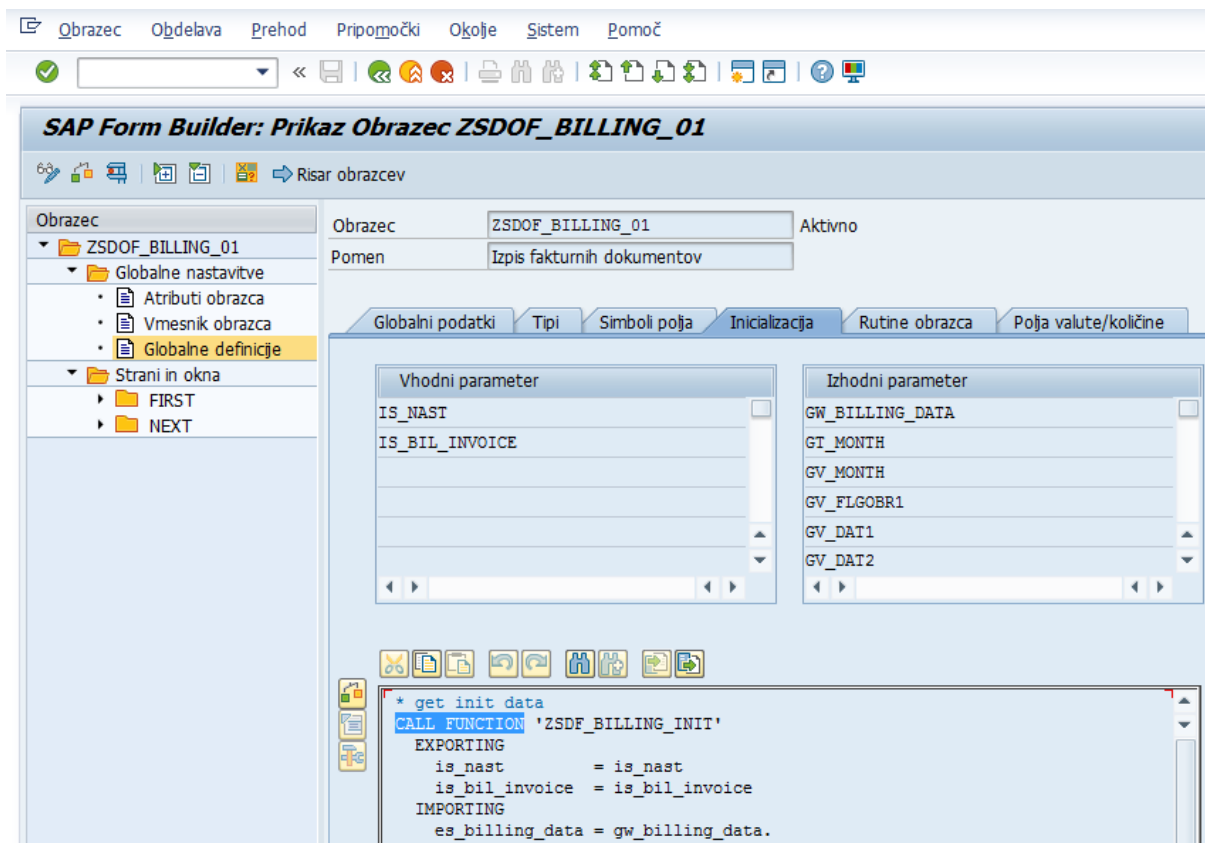
Dokument je sestavljen iz dveh strani, prve strani (FIRST) in naslednje strani (NEXT). Na prvi strani se nahajajo vsi podatki glave, postavke in noge, na vseh naslednjih straneh pa je glava okrnjena. Prva stran FIRST je povezana z naslednjo NEXT, naslednja stran NEXT pa je povezana sama s seboj (slika 10).



Slika 10: Povezava med stranmi

3.2.3 Lastne nastavitve izpisa

Nastavitve izpisa so shranjene v svojih lastnih tabelah in so razdeljene glede na področje izpisa obrazca. Preberejo se ob prvem klicu obrazca v predelu inicializacije (slika 11).



Slika 11: Branje nastavitvev obrazca

Pridobljene nastavitve so shranjene v globalno strukturo obrazca GW_BILLING_DATA in so vidne po celem obrazcu. Struktura GW_BILLING_DATA je sestavljena iz vgnezednih struktur, ki nosijo podatke nastavitvev.

3.2.3.1 Splošne nastavitve

Največji izziv je bil sestaviti logiko nastavitvev in branja cenovnih kalkulacij dokumenta. Cenovna kalkulacije so formule, ki določajo izračune na postavkah in glavi dokumenta. Vsaki vrsti fakturnega dokumenta pripada svoja kalkulacijska shema, ki je sestavljena iz cenovnih kalkulacij. Primer cenovne kalkulacije na postavkah fakturnih dokumentov je na sliki 12.

<input checked="" type="checkbox"/> ZPRO	Redna prodajna cena	204,91803	EUR	1 KOS	204,92	EUR	1 KOS	1 KOS	0,00	<input type="checkbox"/>
	Redna cena - Akcijsk	204,92	EUR	1 KOS	204,92	EUR	1 KOS	1 KOS	0,00	<input type="checkbox"/>
<input checked="" type="checkbox"/> ZIP2	Pogodbeni popust v %	15,000	%		30,74	EUR	0	0	0,00	<input type="checkbox"/>
	DBP avt. RVC	174,18	EUR	1 KOS	174,18	EUR	1 KOS	1 KOS	0,00	<input type="checkbox"/>
	Ocenjena razlika v c	174,18	EUR	1 KOS	174,18	EUR	1 KOS	1 KOS	0,00	<input type="checkbox"/>
	Popusti skupaj	0,00	EUR	1 KOS	0,00	EUR	1 KOS	1 KOS	0,00	<input type="checkbox"/>
	Neto vrednost 20%	174,18	EUR	1 KOS	174,18	EUR	1 KOS	1 KOS	0,00	<input checked="" type="checkbox"/>
	Neto vrednost 22%	174,18	EUR	1 KOS	174,18	EUR	1 KOS	1 KOS	0,00	<input checked="" type="checkbox"/>
	Vrednost brez DDV 20	174,18	EUR	1 KOS	174,18	EUR	1 KOS	1 KOS	0,00	<input checked="" type="checkbox"/>
	Vrednost brez DDV 22	174,18	EUR	1 KOS	174,18	EUR	1 KOS	1 KOS	0,00	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> MWST	Izhodni davek	22,000	%		38,32	EUR	0	0	0,00	<input type="checkbox"/>
	Vrednost z DDV	212,50	EUR	1 KOS	212,50	EUR	1 KOS	1 KOS	0,00	<input checked="" type="checkbox"/>
	Skupaj za plačilo	212,50	EUR	1 KOS	212,50	EUR	1 KOS	1 KOS	0,00	<input type="checkbox"/>
	Skupaj bruto vrednos	212,50	EUR	1 KOS	212,50	EUR	1 KOS	1 KOS	0,00	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> MWST	Izhodni davek	22,000	%		0,00	EUR	0	0	0,00	<input checked="" type="checkbox"/>
	Že plačano	0,00	EUR	1 KOS	0,00	EUR	1 KOS	1 KOS	0,00	<input checked="" type="checkbox"/>
	Razlika za plačilo	212,50	EUR	1 KOS	212,50	EUR	1 KOS	1 KOS	0,00	<input checked="" type="checkbox"/>

Slika 12: Primer cenovne kalkulacije

Izračuni sledijo od vrha navzdol in so nastavljeni v SAP standardnih nastavitvah. Običajno je na vrhu seznama redna prodajna cena, sledijo morebitni popusti, izhodni davek in končni znesek.

Vsak izmed zapisov cenovne kalkulacije vsebuje številko koraka (ki uporabnikom ni vidna) in vrsto pogoja (angl. *condition*). Če zapis vsebuje vrsto pogoja, je podatek zapisan v DB tabeli. Če zapis vrste pogoja ne vsebuje, pomeni, da je izračunan v trenutku, ko smo odprli dokument. Številka koraka je prisotna v vseh primerih.

S kombinacijo številke koraka in vrste pogoja je mogoče izluščiti pravo vrednost, ki naj bo na izpisu prikazana.

Vsebina lastnih nastavitev:

Tipi vrednosti cenovnih pogojev: določen šifrant vrednosti, ki je pozneje uporabljen na izpisih (tabela 1).

Tip vrednosti	Opis vrednosti
00	Ne izpiši
01	Cena brez DDV
03	Popust
05	Manipul. stroški
08	Vrednost brez DDV
15	Akcijski popusti skupaj
16	Internetni popust skupaj
17	Popusti skupaj
18	Vrednost blaga in storitev
19	Vrednost brez DDV
20	Vrednost DDV - #1
21	Zaokroževanje
22	Skupaj za plačilo
25	Vrednost DDV - #2
26	Znesek predplačila
27	Bruto vrednost
28	Že plačano
29	Razlika za plačilo

Tabela 1: Tipi vrednosti cenovnih pogojev

Določitev cenovne kalkulacije: določa povezavo med tipom vrednosti, cenovno kalkulacijo in vrsto računa (tabela 2):

Kalkulacijska shema	Tip računa	Številka koraka	Pogoj	Tip vrednosti
ZTS001	ZSZV	002	ZPR0	01
ZTS001	ZSZV	002	ZPR0	08
ZTS001	ZSZV	915	MWST	00
ZTS001	ZSZV	932	MWSI	00
ZTS002	ZA1	900	ZTPP	26
ZTS002	ZA1	915	MWSI	20
ZTS002	ZA1	934	NETW	01
ZTS002	ZA1	934	NETW	08
ZTS002	ZA1	935		28

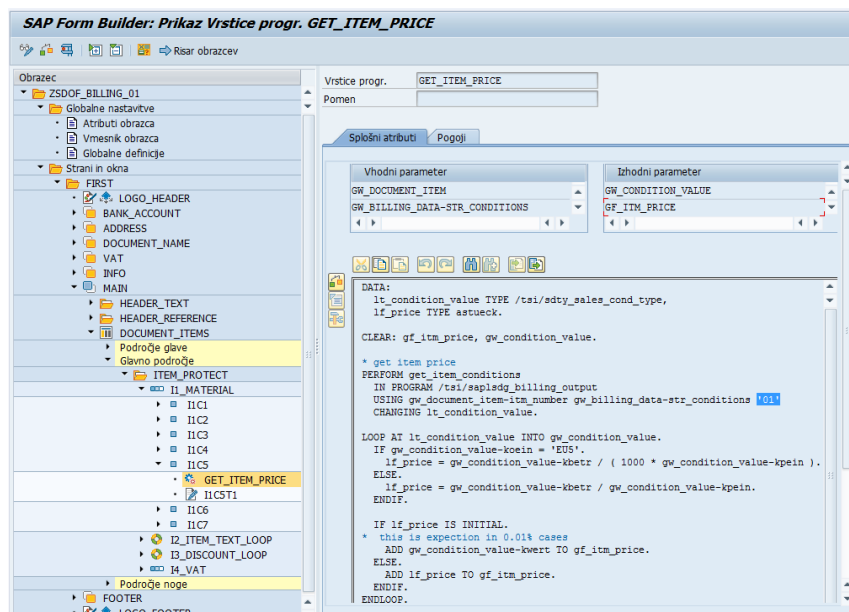
Tabela 2: Določitev cenovne kalkulacije

Branje cenovnih kalkulacij poteka s funkcijskim modulom ZSDF_BILLING_CONDITIONS. Funkcijski modul vrne cenovno kalkulacijo postavke ali glave dokumenta (tabela 3):

CALC_TYPE	STUNR	KSCHL	KSCHL_DESC	KBETR	KOEIN	KPEIN	KMEIN	KWERT	WAERK
	1	ZPP0	Plačilni pogoji kupca	0,00	%	0		0,00	EUR
01	2	ZPR0	Redna prodajna cena	245.901,64	EU5	1	ST	245,90	EUR
18	6		Vrednost brez popust	245,90	EUR	1	ST	245,90	EUR
	14		Internetna cena	245,90	EUR	1	ST	245,90	EUR
	100		Prodajna vrednost	245,90	EUR	1	ST	245,90	EUR
03	110	ZTP2	Pogodbeni popust v %	-15,00	%	0		-36,89	EUR
17	300		Popusti skupaj	-36,89	EUR	1	ST	-36,89	EUR
16	310		Internetni popust sk	0,00	EUR	1	ST	0,00	EUR
15	320		Akcijski popust skup	0,00	EUR	1	ST	0,00	EUR
	360		Neto vrednost	209,01	EUR	1	ST	209,01	EUR
19	370		Vrednost brez DDV	209,01	EUR	1	ST	209,01	EUR
20	915	MWST	Izhodni davek	22,00	%	0		45,98	EUR
	916		Vrednost z DDV	254,99	EUR	1	ST	254,99	EUR
22	920		Skupaj za plačilo	254,99	EUR	1	ST	254,99	EUR
	925		Skupaj bruto vrednos	254,99	EUR	1	ST	254,99	EUR
25	932	MWSI	Izhodni davek	22,00	%	0		0,00	EUR
28	933		Že plačano	0,00	EUR	1	ST	0,00	EUR
29	934		Razlika za plačilo	254,99	EUR	1	ST	254,99	EUR
	940	VPRS	Strošek	205,01	EUR	1	ST	205,01	EUR
08	2	ZPR0	Redna prodajna cena	245.901,64	EU5	1	ST	245,90	EUR

Tabela 3: Rezultat funkcijskega modula – cenovna kalkulacija postavke

Na *smartform* obrazcu, v predelu postavk, sistem prebere posamezne vrednosti. Slika 13 prikazuje branje cene brez DDV (tip vrednosti 01) iz globalne strukture.



Slika 13: Primer branja cenovne kalkulacije

3.2.3.2 Nastavitve glave računa

Naziv dokumenta: glede na vrsto dokumenta je dinamično določen naziv dokumenta. Ker je lahko obrazec vzdrževan v različnih jezikih, je potrebno zraven vrste dokumenta določiti tudi jezik (tabela 4).

Tip računa	Jezik izpisa	Naziv dokumenta
Z7	SL	RAČUN
Z8	EN	INVOICE
Z8	SL	RAČUN
Z9	EN	INVOICE
Z9	SL	RAČUN
ZA	EN	INVOICE
ZA	SL	RAČUN
ZA1	EN	ADVANCE PAYMENT
ZA1	SL	AVANSNI RAČUN
ZA2	EN	ADVANCE PAYMENT
ZA2	SL	AVANSNI RAČUN
ZAF2	EN	INVOICE
ZAF2	SL	RAČUN
ZAFS	EN	INVOICE
ZAFS	SL	RAČUN
ZAG2	EN	CREDIT NOTE
ZAG2	SL	DOBROPIS
ZAP	EN	INVOICE
ZAP	SL	RAČUN

Tabela 4: Primer nastavitve naziva dokumenta

Določitev tekstov glave: teksti glave so lahko vnaprej določeni z modulom teksta, lahko pa se berejo iz samega dokumenta fakture.

Tekste na dokumentu fakture uporabnik sam vpiše. Vsebinski svetovalec si lahko sam nastavi kateri teksti se bodo v glavi prikazali (tabela 5).

Tip	Št.	Naziv teksta	Objekt	ID	ID teksta	Tekst pred	Tekst za
ZTMA	1		VBBK	Z011	DC_00		ZSDX_BLANK_01
ZTMA	2		VBBK	Z005	DC_00		
ZTMA	3		VBBK	Z009	DC_00		
*	1		VBBK	ZT01	DC_00		
Z3	1	ZSDX_ESHOP			MO_00		ZSDX_BLANK_01
ZA	1		VBBK	ZT01	DC_00		ZSDX_BLANK_01
ZA	2	ZSDX_PERSONELL_AV			MO_00		ZSDX_BLANK_01
ZAF2	1		VBBK	0001	DC_00		ZSDX_BLANK_01
ZAF2	2	ZSDX_PERSONELL_AV			MO_00		ZSDX_BLANK_01
ZAFS	1		VBBK	0001	DC_00		ZSDX_BLANK_01
ZAFS	2	ZSDX_PERSONELL_AV			MO_00		ZSDX_BLANK_01
ZAG2	1		VBBK	0001	DC_00		ZSDX_BLANK_01
ZAG2	2	ZSDX_PERSONELL_AV			MO_00		ZSDX_BLANK_01
ZAP	1		VBBK	0001	DC_00		ZSDX_BLANK_01
ZAP	2	ZSDX_PERSONELL_AV			MO_00		ZSDX_BLANK_01
ZE04	1	ZSDX_EKC			MO_00		ZSDX_BLANK_01

Tabela 5: Primer določitev tekstov glave

Če sta objekt in ID vpisana, pomeni, da sistem bere tekste iz glave dokumentov, naziv teksta ni pomemben. Če je naziv teksta vpisan, sistem bere tekst iz modula teksta.

Datumi v glavi: prikaz datumov se razlikuje glede na vrsto dokumenta (tabela 6).

Tip računa	Jezik	Opis datuma	Tip datuma
ZISB	EN	Debit memo, &CITY&:	B
ZISB	SL	Izdaja bremepisa, &CITY&:	B
ZISD	EN	Credit memo, &CITY&:	B
ZISD	SL	Izdaja dobropisa, &CITY&:	B
ZISF	EN	Invoicing, &CITY&:	B
ZISF	SL	Izdaja računa, &CITY&:	B
ZKA1	EN	Skopje:	D

Tabela 6: Primer nastavitve prikaza datuma

Recimo pri izpisu fakture mora biti datum oblike »Datum računa, DD.MM.LLLL«, pri izpisu dobropisa »Datum dobropisa, DD.MM.LLLL«.

Tip datuma določa pomen datuma oz. od kje je datum prebran:

- A Datum placila - PAYDATE
- B Datum ko je bil dodan zapis - BIL_EDATE
- C Datum fakturiranja - BIL_DATE
- D Opis plačilnega pogoja, brez datuma

Določitev TRR v glavi: TRR se izpisuje glede na šifro podjetja ki izstavlja račun (tabela 7). Če šifra podjetja v nastavitvah ni določena, sistem vzame zapis pod ključem *šifra podjetja* = *

Tip računa	Šifra podjetja	Bančni račun
*	*	SI56 0431 2030 1033 713 NKBM, SWIFT: KBMASI2X
*	*	SI56 0292 2601 1098 965 NLB, SWIFT: LJBASI2X
*	*	SI56 0310 0140 6255 420 SKB, SWIFT: SKBASI2X
*	*	SI56 0510 0811 3087 356 ABANKA, SWIFT: ABANSI2X
*	*	SI56 2900 0100 1843 306 UNICREDIT: SWIFT - BACXSI22
*	1400	
*	1500	
*	1600	

Tabela 7: Primer nastavitve izpisa bančnega računa

3.2.3.3 Nastavitve postavke računa

Določitev tekstov postavke: teksti postavke so lahko vnaprej določeni teksti z modulom teksta ali uporabniško definirani teksti na postavki dokumenta. Struktura nastavitve je ista kot pri tekstu glave. Vsebina nastavitve se razlikuje v delu uporabniško vpisanih tekstov. Uporabniško vpisani teksti se vodijo pod *Objekt* = VBPP.

3.2.3.4 Nastavitve noge računa

Določitev končne kalkulacije: določen je vrstni red izpisa končnih izračunov postavk (tabela 8).

Tip računa	Zap. št.	Tip vrednosti
ZTMA	1	18
ZTMA	2	17
ZTMA	3	19
ZTMA	4	20
ZTMA	5	22
ZTMA	6	28
ZTMA	7	29
*	1	18
*	2	17
*	3	19
*	4	20
*	5	21
*	6	22

Tabela 8: Primer končni izračuni

Nazivi končne kalkulacije: nazivi so odvisni od jezika prikazanega obrazca, zato je v tabelo dodan ključ jezika (tabela 9):

Tip računa	Jezik	Tip vrednosti	Tekst kalkulacije
ZTMA	EN	21	Rounding:
ZTMA	SL	21	Zaokroževanje:
*	EN	17	DISCOUNT:
*	EN	18	VALUE:
*	EN	19	VALUE WITHOUT VAT:
*	EN	20	VALUE &VAT& VAT:
*	EN	22	TOTAL AMOUNT &CURR&:
*	EN	28	ALREADY PAID:
*	EN	29	FOR PAYMENT :
*	SL	16	VREDNOST POPUSTA:
*	SL	17	VREDNOST POPUSTA:
*	SL	18	VREDNOST BLAGA IN STORITEV:
*	SL	19	VREDNOST BREZ DDV:
*	SL	20	VREDNOST &VAT& DDV:
*	SL	22	SKUPAJ &CURR&:
*	SL	28	ŽE PLAČANO:
*	SL	29	RAZLIKA ZA PLAČILO:

Tabela 9: Primer nazivov končnih izračunov

Določitev tekstov noge: v nogi se nahajajo zakonsko določene klavzule in morebitna obvestila prejemniku računa (tabela 10). Struktura nastavitev je ista kot pri tekstih glave in tekstih postavke. Nastavitve vsebujejo tudi slike žigov in morebitne podpise odgovornih oseb.

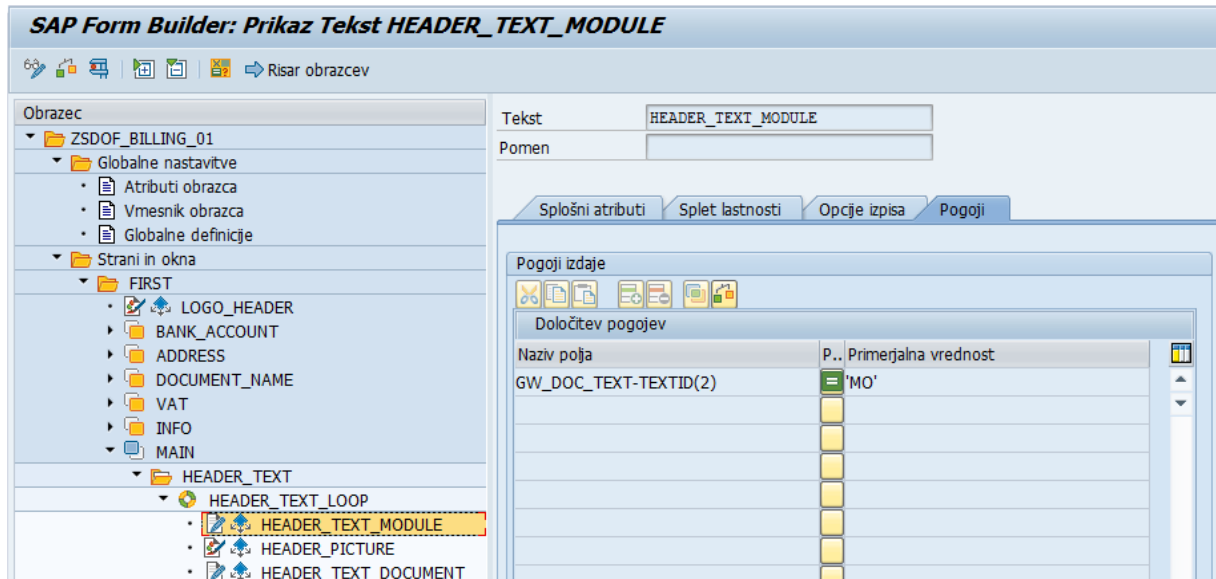
Tip računa	Št.	Naziv teksta	Objekt	ID	ID teksta	Tekst pred	Tekst za
ZTMA	1	ZSDX_PREPAYMENT_SPECIFIC			MO_04		
ZTMA	2	ZSDX_PAYMENT_TSM			MO_00	ZSDX_BLANK_01	
ZTMA	3	ZSDX_76A_ZDDV1			MO_01	ZSDX_BLANK_01	
ZTMA	4	ZSDX_76A_ZDDV1_22			MO_01	ZSDX_BLANK_01	
ZTMA	5	ZSDX_ISSUER_TSM			MO_00	ZSDX_BLANK_01	
ZTMA	6	ZSDX_PODPIS_IN_ZIG_2013			GR_00		
*	1		VBBK	ZT02	DC_00		
*	2	ZSDX_PREPAYMENT_SPECIFIC			MO_04		
*	3	ZSDX_PAYMENT			MO_00		
*	4	ZSDX_76A_ZDDV1			MO_01		
*	5	ZSDX_76A_ZDDV1_22			MO_01		
*	6	ZTS_ZIG_PODPIS_2013			GR_00		

Tabela 10: Primer določitve tekstov noge

Nastavitve glave, postavk in noge so ločene na tri dele, glede na nastavitev ID teksta:

- teksti modula MO,
- teksti dokumenta DC in
- slike GR

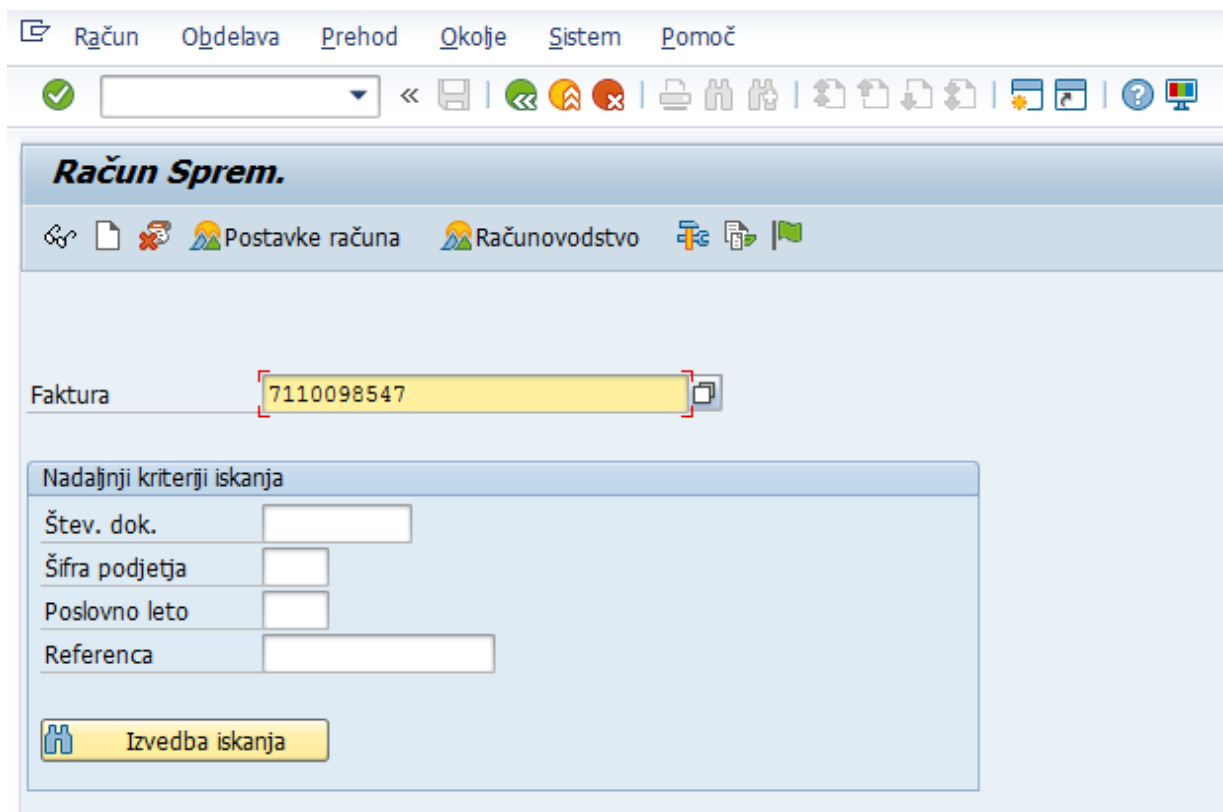
V *smartform* obrazcu se posamezni deli (teksti modula, teksti dokumenta, slike) ločeno kličejo (slika 14).



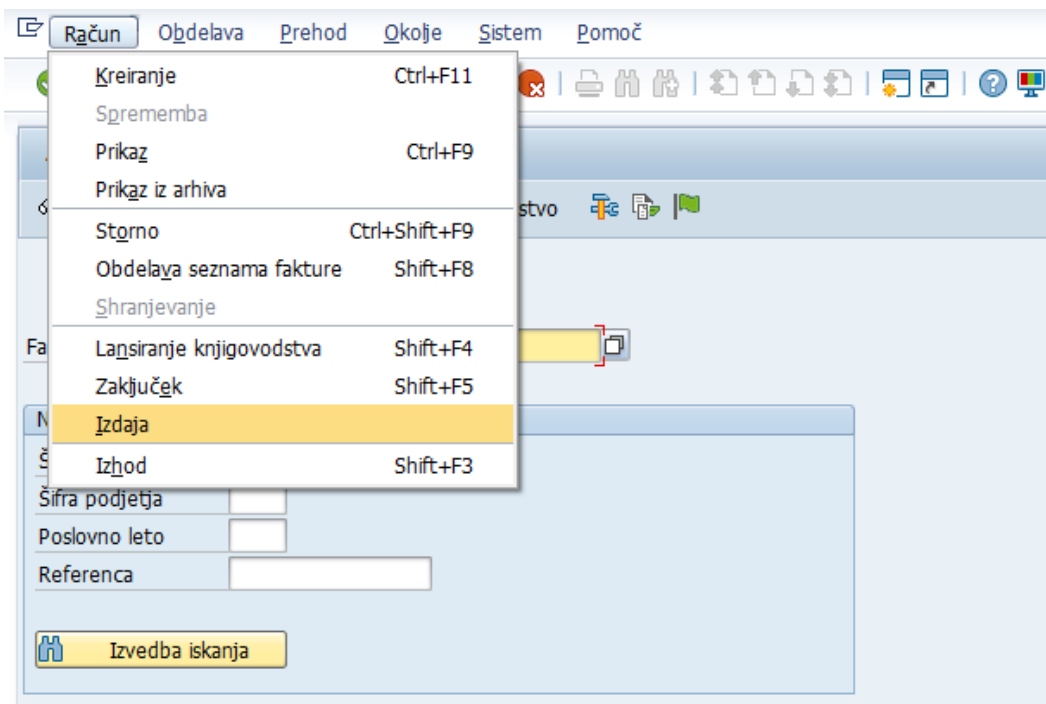
Slika 14: Klicanje modul teksta, teksta dokumenta in slik

3.3 Primer izpisa fakture

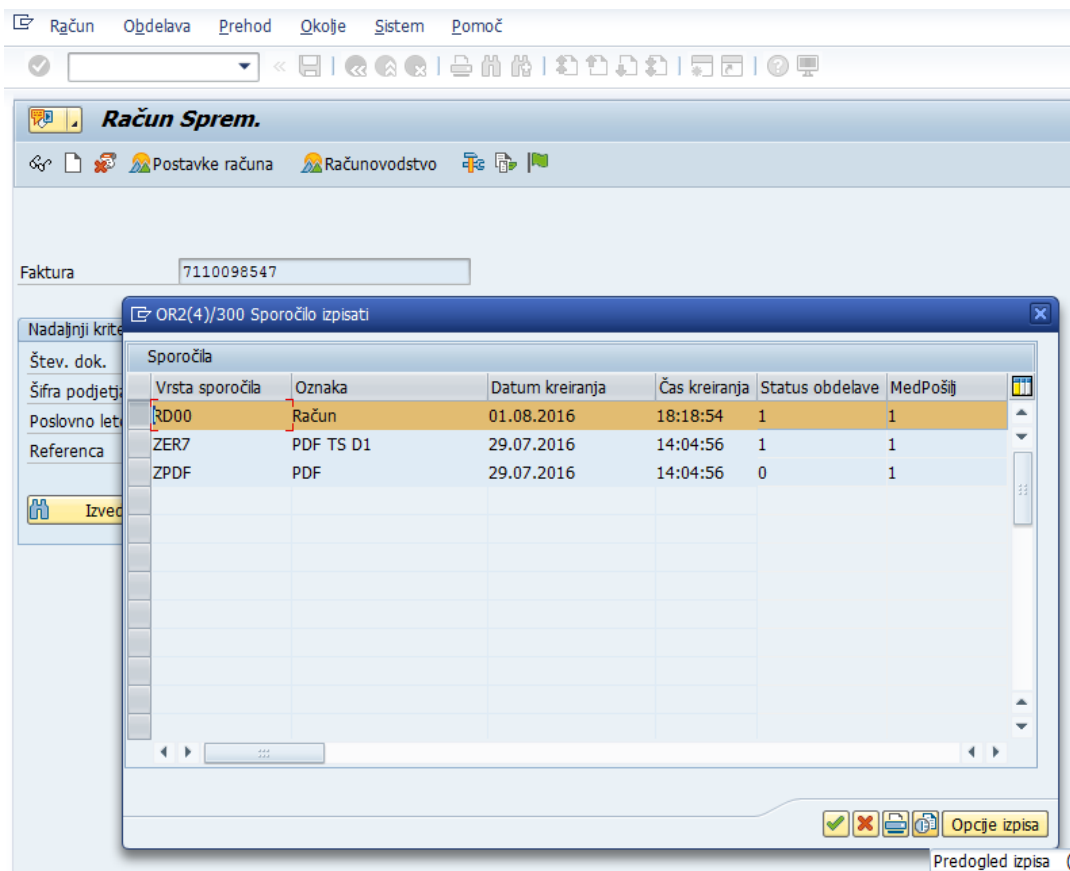
Izpis fakture se izvaja v transakciji **VF03** za prikaz fakturnega dokumenta oz. v transakciji **VF02** za spremembo dokumenta (slika 15, slika 16, slika 17, slika 18).



Slika 15: Transakcija za spremembo fakturnega dokumenta



Slika 16: Izbira izdaje izhoda fakturnega dokumenta



Slika 17: Izbira izhoda na fakturnem dokumentu

SI56 [REDACTED] NKBM, SWIFT: KBMASI2X
 SI56 [REDACTED] NLB, SWIFT: LJBAS12X
 SI56 [REDACTED] SKB, SWIFT: SKBAS12X
 SI56 [REDACTED] ABANKA, SWIFT: ABANSI2X
 SI56 [REDACTED] UNICREDIT, SWIFT: BACXSI22

[REDACTED]
 [REDACTED]

2000 MARIBOR

ID za DDV: SI2 [REDACTED]
 Zavezanec za DDV: DA
 Kupec: 34

RAČUN št.: TS3-D1-7110098547

Ljubljana, 29. 7. 2016
 Datum valute: 28. 8. 2016
 Datum dobave blaga/opravljenе storitve: 29. 7. 2016

Vezni dokument: TS3-D5-7210001277

Artikel	Oznaka PO	EM	Količina	Cena v EUR	Popust	Vrednost EUR
[REDACTED] vzdrževanje	50005416	KOS	5	200,00000		1.000,00
DDV 22,00%						220,00
VREDNOST BLAGA IN STORITEV:						1.000,00
VREDNOST BREZ DDV:						1.000,00
VREDNOST 22,00% DDV:						220,00
SKUPAJ EUR:						1.220,00
ŽE PLAČANO:						999,00
RAZLIKA ZA PLAČILO (Osnova DDV 172,38, 22,00% DDV 48,62):						221,00

Prejemnik blaga: [REDACTED]
 [REDACTED], 2000 MARIBOR

Št. naročila 3064205
 Št. kup. naročila

Specifikacija koriščenja predplačila:

Št. avansnega računa	Datum	Vrednost predplačila	Prer.st.DDV v %	Vračunan DDV	Predplačilo brez DDV
7210001277	22.07.2016	999,00			999,00

PROSIMO, DA ZNESEK NAKAŽETE NA RAČUN ŠT. SI56 [REDACTED], PRI TEM OBVEZNO NAVEDITE SKLIC: SI12 0071100985476

V PRIMERU NEPRAVOČASNEGA PLAČILA ZARAČUNAVAMO ZAKONSKE ZAMUDNE OBRESTI.

Slika 18: Predogled izpisa računa

3.4 Prednosti in slabosti rešitve prednastavljenih izpisov

Prednosti:

Glavna prednost rešitve je, da v večini primerov ni potreben poseg ABAP programerja. Vsebinski svetovalec lahko sam, med procesom nastavljanja sistema, poskrbi tudi za izpise. Prednost rešitve je tudi ta, da je za sorodne izpise razvit isti *smartform* obrazec.

Splošna praksa je, da se je v posameznih *smartform* izpisih potrebno prilagajati nastavitvam sistema. Nastavitve v izpisih so »zapečene«, primer: glede na vrsto dokumenta se na izpisu prikazuje ustrezen naziv dokumenta. V izpisu so tako kar v kodi našteje vse uporabljene vrste dokumentov in njihovi pripadajoči nazivi. Izpis na drugem sistemu ne bo deloval, ker so po vsej verjetnosti določeni drugi ključi za vrste dokumentov. S prednastavljenimi izpisi teh problemov ni več, ker vsebinski uvajalec poskrbi za pravilne vnose nastavitvev.

Slabosti:

Največja slabost je prednastavljena oblika. Oblika zadostuje zakonskim zahtevam. Stranka se mora odločiti ali bo investirala v razvoj svoje lastne oblike oz. ali se bo sprijaznila z prednastavljeno obliko in s tem privarčevala.

Spremembe na obrazcu, ki jih brez ABAP programerja ni možno rešiti:

- sprememba fonta,
- sprememba barve,
- sprememba pozicije oken,
- določitev širine stolpcev tabele, recimo pri izpisu postavk,
- sprememba vrstnega reda stolpcev in
- sprememba resolucije slik

Priložnosti:

Ponudba prednastavljenih izpisov pomeni konkurenčno prednost pred ostalimi ponudniki informacijskih rešitev. To pomeni, da ima svetovalno podjetje večje možnosti da bo izbrano za izvajalca storitev, ko potencialna stranka izda razpis za izbiro ponudnika novega informacijskega sistema.

Nevarnosti:

Ker je rešitev integrirana v SAP standardno okolje obstaja možnost, da bo integracija nehala delovati ob namestitvi SAP podpornih paketov (angl. *Support Packages*). V ta namen je potrebno rešitev testirati po vsaki SAP posodobitvi in v primeru nedelovanja prilagoditi *ABAP* kodo tako, da je prilagojena stari in novi verziji SAP integracije.

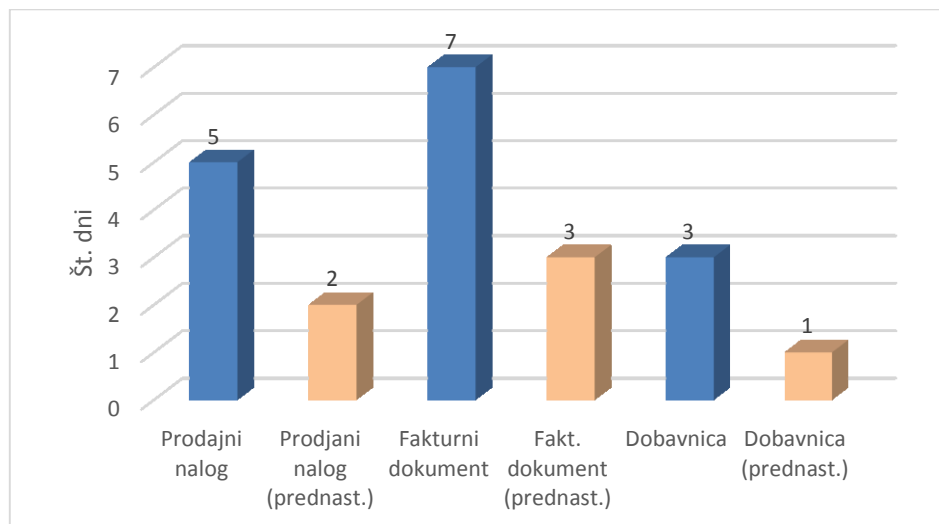
4 Sklepne ugotovitve

Običajna praksa pri razvoju izpisov je, da se programer prilagaja nastavitvam sistema. Glede na vrsto dokumenta in nastavitve sistema ABAP programer nastavi tekste obrazca, naloži morebitne logotipe in določi katere vrednosti se bodo na obrazcu pojavile. Vse nastavitve so fiksno zapisane na obrazcu, tako da ga ni mogoče uporabiti pri drugih implementacijah.

Namen diplomskega dela je zmanjšanje časa in stroškov projekta v delu implementacije izpisov. Ideja je, da s prednastavljenimi izpisi, ki zadoščajo zakonskim potrebam, pokrijemo vse potrebe podjetja in skrajšamo čas implementacije.

V ta namen so bili razviti prednastavljeni izpisi, ki se jih lahko prenaša iz sistema na sistem in pripravi do delovanja samo z nastavitvami brez posega programerja. V diplomskem delu smo opisali razvoj in nastavitve prednastavljenega izpisa fakturnega dokumenta v SD modulu prodaje.

V običajnih razmerah je ocenjen čas razvoja fakturnih izpisov 7 dni, izpisov prodajnega naloga 5 dni in izpisov dobavnih dokumentov 3 dni, kar nanese skupaj 15 dni programiranja. Z našo rešitvijo smo dosegli krajši čas implementacije izpisov za 60%, iz 15 dni na 6 dni (slika 19).



Slika 19: Primerjava trajanja implementacije prodajnih izpisov

V **prvi fazi** bo nadaljni razvoj rešitve usmerjen v optimizacijo ABAP kode. Pri izpisih z več postavkami (1000 ali več) so odzivni časi slabi. To se dogaja zato, ker sistem med izvajanjem izpisa neprestano preverja nastavitve z `select single` stavkom. Logiko branja nastavitvev je potrebno prilagoditi tako, da se ob začetku izvajanja izpisa najprej preberejo vse nastavitve iz podatkovne baze v interno tabelo, nato pa se iz interne tabele bere posamezne zapise.

V **drugi fazi** bomo razširili funkcionalnost tudi na druga SAP orodja za kreiranje izpisov (recimo *Adobe Lifecycle Designer*). Nastavitve bodo ostale iste kot pri SAP *Smartforms*, spremenili se bodo le podporni programi za obdelavo izpisa.

5 Viri in literatura

[1] SAP AG. (2016). *ASAP Methodology for Implementation*. [Online].

Dosegljivo: <https://support.sap.com/support-programs-services/methodologies/implement-sap/asap-implementation.html>

[2] SAP AG. (2016). *SAP history*. [Online]

Dosegljivo: <http://go.sap.com/corporate/en/company/history.html>

[3] Wikipedia. (2016). *SAP R/3*. [Online]. Dosegljivo: https://en.wikipedia.org/wiki/SAP_R/3

[4] SAP AG. (2016). *Overview of NetWeaver AS ABAP*. [Online].

Dosegljivo: https://help.sap.com/saphelp_nw70/helpdata/en/fc/eb2e97358411d1829f0000e829fbfe/frameset.htm

[5] Arun Krishnamoorthy. (2016). *Flow of an ABAP program*. [Online].

Dosegljivo: <http://www.teamabap.com/2014/05/flow-of-abap-program.html>

[6] SAP AG. (2016). *ABAP dictionary*. [Online].

Dosegljivo: https://help.sap.com/saphelp_nw73ehp1/helpdata/en/cf/21ea0b446011d189700000e8322d00/content.htm?current_toc=/en/4f/991f82446d11d189700000e8322d00/plain.htm&show_children=true

[7] SAP AG. (2016). *ABAP Workbench Tools*. [Online].

Dosegljivo: https://help.sap.com/saphelp_nw73ehp1/helpdata/en/ef/d94b78ebf811d295b100a0c94260a5/frameset.htm

[8] SAP AG. (2016). *Development Namespaces*. [Online].

Dosegljivo: <https://support.sap.com/namespaces>

[9] SAP AG. (2016). *Output Determination Using Condition Technique*. [Online].

Dosegljivo: https://wiki.scn.sap.com/wiki/display/ERPLO/output+determination+using+condition+technique?original_fqdn=wiki.sdn.sap.com

[10] SAP AG. (2016). *Smartforms*. [Online].

Dosegljivo:<https://wiki.scn.sap.com/wiki/display/ABAP/Smart+Forms>

[11] SAP AG. (2016). *Node Types: Overview*. [Online].

Dosegljivo:http://help.sap.com/saphelp_nw70/helpdata/en/d2/69a148cd8b11d3b563006094192fe3/frameset.htm

[12] SAP AG. (2016). *Table maintenance generator*. [Online].

Dosegljivo:<https://wiki.scn.sap.com/wiki/display/ABAP/TABLE+MAINTENANCE+GENERATOR+and+ITS+EVENTS>